

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.78

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: _____ Діалогова підсистема інтелектуального агента

Виконав: студент другого курсу, групи ІК-81мп
(шифр групи)

_____ Безпалько Олександр Сергійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н. Олійник В.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант норм.контроль _____ к.т.н., доц. Пасько В.П. _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ І.Р. Пархомей
(підпис)

«__» _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Безпалько Олександр Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Діалогова підсистема інтелектуального агента» _____

науковий керівник дисертації Олійник Володимир Валентинович, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28» жовтня 2019 р. №3770-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження - процес автоматизованої обробки вільної мови _____

4. Предмет дослідження - діалогова підсистема інтелектуального агента

5. Перелік завдань, які потрібно розробити - аналіз проблематики, аналіз існуючих рішень, аналіз підходів до реалізації, вибір технологій для рішення, розробка додатку, дослідження ефективності розробленого додатку.

6. Орієнтовний перелік ілюстративного матеріалу – шість плаката.

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
НК	Пасько В.П., доцент		
Перевірка на співпадіння	Лісовиченко О.І., доцент		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	14.09.2019 р.	
2	Постановка задачі	16.09.2019 р.	
3	Вибір технологій розробки	21.09.2019 р.	
4	Розробка програмного забезпечення	26.09.2019 р.	
5	Тестування системи	16.10.2019 р.	
6	Розробка рекомендацій щодо впровадження системи	02.11.2019 р.	
7	Аналіз стартап-проекту	11.11.2019 р.	
8	Висновки	16.11.2019 р.	

Студент

(підпис)

Безпалько О.С.
(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Олійник В.В.
(ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто існуючі методи та підходи до проектування діалогових підсистем інтелектуальних агентів, показано основні особливості існуючих рішень проблеми, їх переваги та недоліки. Проведено дослідження сучасних методів їх реалізації.

Розроблено діалогову підсистему інтелектуального агента, що надає можливості розпізнавання вільної мови. Система є унікальною у своєму роді так як працює на основі української мови, для якої не існує жодного відомого рішення. Дана система може бути використана практично у будь-якій галузі і надає широкі можливості автоматизації взаємодії з користувачами чи клієнтами. Дозволяє значно покращити рівень комфорту при отриманні послуг, або довідкової інформації, покращує точність на швидкість за рахунок особливої реалізації, дає змогу зекономити ресурси на розробку.

Ключові слова: інтелектуальний агент, природнє розпізнавання мови, машинне навчання, штучний інтелект.

Розмір пояснювальної записки – 89 аркушів, містить 40 ілюстрацій, 22 таблиць, 6 додатків.

ABSTRACT

The paper examines the existing methods and approaches to designing dialogue subsystems of intellectual agents, shows the main features of existing solutions to the problem, their advantages and disadvantages. Modern methods of their implementation are investigated.

An intelligent agent dialog subsystem has been developed to enable free speech recognition. The system is unique in that it operates based on the Ukrainian language, for which there is no known solution. This system can be used in almost any industry and offers many opportunities to automate interaction with users or customers. It can significantly improve the comfort level of receiving services or background information, improve precision and speed through special implementation, save on development resources.

Keywords: intellectual agent, natural language recognition, machine learning, artificial intelligence.

Explanatory note size – 89 pages, contains 40 illustrations, 22 tables, 6 applications.

**Пояснювальна записка
до магістерської дисертації**

на тему: *Діалогова підсистема інтелектуального агента*

Київ – 2019 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	11
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	13
1.1 Діалогові системи	13
1.2 Аналіз особливостей.....	14
1.3 Огляд існуючих рішень.....	19
1.3.1 Огляд інтелектуальних помічників на основі команд	20
1.3.1.1 Mondly	20
1.3.1.2 Buoy Health Chatbot.....	21
1.3.1.3 Realbot Chatbot	22
1.3.2 Огляд інтелектуальних помічників на основі нейронних мереж	23
1.3.2.1 Casper.....	23
1.3.2.2 Roof AI	24
1.3.2.3 Mya Chatbot.....	25
1.3.3 Огляд особливих реалізацій інтелектуальних помічників.....	26
1.3.3.1 Visual Dialog	26
1.3.3.2 Hipmunk.....	27
1.3.3.3 UNICEF	27
1.4 Аналіз підходів до побудови інтелектуальних агентів.....	28
1.5 Постановка задачі	31
Висновки до розділу	31
РОЗДІЛ 2. ІНФОРМАЦІЙНА МОДЕЛЬ	32
2.1 Функціональна модель	32
2.1.1 NLU.....	33
2.1.2. Core	34
2.1.3 Принцип функціонування	35
2.2 Підготовка даних	39
2.2.1 ВЕСУМ.....	39
2.2.2 MITIE.....	39
2.2.3 Dlib.....	40
2.2.4 Word2vec	42

2.3	Нейронна мережа.....	46
2.3.1	Recurrent Neural Networks	46
2.3.2	RNN на основі LSTM.....	48
2.3.3	Архітектура LSTM.....	49
2.4	Процес навчання	53
	Висновки до розділу	54
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....		55
3.1	Архітектура програмного забезпечення	55
3.2	Взаємодія з користувачем	56
3.3	Алгоритм роботи.....	57
3.4	Вибір засобів для розробки	61
3.5	Вимоги до технічного та програмного забезпечення.....	62
3.6	Результати роботи	63
	Висновки до розділу	65
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ		66
4.1	Опис ідеї проекту	67
4.2	Технологічний аудит ідеї проекту.....	71
4.3	Аналіз ринкових можливостей запуску стартап-проєкту.....	72
4.4	Розроблення ринкової стратегії проекту	80
4.5	Розроблення маркетингової програми стартап-проєкту.....	83
	Висновки до розділу	86
ВИСНОВКИ.....		87
ПЕРЕЛІК ПОСИЛАНЬ		89
ДОДАТКИ.....		90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AL – Artificial intelligence;

ML – Machine Learning;

NLP – Natural language processing;

NER – Named-entity recognition;

LSTM – Long short-term memory;

GRU – Gated Recurrent Units;

CBOW – Common Bag Of Words;

RNN – Recurrent Neural Networks;

LR-MVL – Low rank multi-view learning;

CCA – Canonical correlation analysis;

DWE – Distributional word embeddings;

SSVM – Structural support vector machines;

UNICEF – The United Nations Children's Fund;

HTML – HyperText Markup Language;

API – Application programming interface;

SQL – Structured Query Language;

UML – Unified Modeling Language;

API – Application programming interface;

IDE – Integrated Development Environment;

АШНС – Архітектура штучної нейронної сітки;

ЕОМ – Електронно-обчислювальна машина;

МНШНС – Методи навчання штучних нейронних сіток;

НС – Нейронна сітка;

ПЗ – Програмне забезпечення;

ІТ – Інформаційні технології;

НТ – Нейротехнологія;

ШІ – Штучний інтелект;

ШН – Штучний нейрон;

ШНС – Штучна нейронна сітка;

ШНМ – Штучна нейромережа;

БД – База даних;

СУБД – Система управління базами даних;

ОС – Операційна система.

ВСТУП

Сьогодні, в епоху сучасних інформаційних технологій, коли у всіх видах підприємницької діяльності та у сферах послуг намагаються досягти максимального комфорту при роботі з клієнтами і автоматизувати цей процес взаємодії, все більш актуальними стають чат-боти.

Чат-боти створюють, щоб позбавитись проблем, з яким стикаються сьогодні галузі. Мета чат-ботів – підтримка та розширення ділових команд у стосунках із клієнтами. Він може жити у будь-яких великих додатках для чату, таких як Facebook Messenger, Slack, Telegram, тощо.

Чат-боти можуть здатися футуристичним поняттям, але згідно зі статистикою глобального веб-індексу говориться, що 75% користувачів Інтернету користуються однією або кількома платформами обміну повідомленнями. Хоча дослідження показують, що кожен користувач використовує в середньому 24 додатки на місяць, при цьому 80% часу припадає лише на 5 програм. Безперечно, серед них є Facebook Messenger, Snapchat, Whatsapp, WeChat і т. д. Це означає, що ви навряд чи створення власного додатку є необхідним, але у вас все ще є високі шанси інтегрувати ваш чат з однією з цих платформ.

Хіба не було б чудово, якби хтось міг полегшити вам життя, допомагаючи 24 на 7. Чат-боти можуть робити саме це!

Для бізнесу стало необхідним вирішувати запити та проблеми покупців, щоб забезпечити лояльність споживачів разом із створенням бренду. І так само, як і в попередні часи, людина намагалася скористатися машинами, щоб позбутися людських обмежень, ця тенденція продовжується і зараз. Цього разу революція стала галуззю обслуговування клієнтів, і нововведенням, що відповідає за це, є чат-бот. Чат-боти вважаються майбутнім обслуговування клієнтів та управління.

Комп'ютерні програми, які можуть вести справжні розмови, називаються чат-ботами. Інтерфейс чату дозволяє спілкуватися з користувачами. Інтелектуальних

агентів можна використовувати майже з усіма популярними програмами для обміну повідомленнями. Цим ботам можна також надати різних особистостей. Вони можуть розуміти письмовий та розмовний текст та інтерпретувати його значення. Потім вони можуть шукати відповідну інформацію та доставляти її користувачеві.

Чат-боти мають широкий спектр застосувань. Оскільки дані показують, що все більше і більше людей спілкуються, а не телефонують, буде потреба у використанні чатів декількома галузями . Причиною такої вибухової популярності є зростання мобільних месенджерів та розширення технології штучного інтелекту. Це суттєво поліпшить їхні доходи та задоволеність клієнтів. Оскільки технології рухаються швидше, ніж будь-коли раніше, створити чат із такою кількістю платформ стало дуже просто . Можна впевнено сказати, що настає революція чат-ботів, і ми будемо готові прийняти та інтегрувати її в наш бізнес.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Діалогові системи

Область, в якій розглядається проблема, є автоматизація взаємодії з користувачем, за допомогою програмних засобів, при отримання ним певної інформації або ряду послуг.

Зараз діалогові системи досить розумні, щоб зрозуміти закономірності та надіслати відповідь, але для цього вони пройшли довгий шлях. Прості чат-боти змогли зіставити текстовий рядок і запропонувати відповідь лише тоді, коли буде знайдена точна відповідність. Коли ми сказали, що чат-боти пройшли довгий шлях, ми насправді це мали на увазі. У наш час почали розроблятися технології для створення інтелектуальних агентів із застосуванням нейронних мереж. Використовується крива навчання, що працює на основі штучного інтелекту, і це веде до того, що вони отримують велике значення.

Об'єктом дослідження є процес автоматизованої обробки вільної мови. Нейронні мережі, при створенні чат-ботів, почали застосовуватися порівняно не давно. А у відношенні розуміння вільної мови тільки з'являться перші спроби вирішити цю задачу. Це дещо простіше реалізувати при використанні англійської мови, так як там кількість словоформ набагато менша у порівнянні з іншими. Створення чогось подібного на слов'янській групі мов викликає ряд складнощів. Існує деякий ряд рішень на російській мові, так як вона є найбільш популярною серед зазначеної мовної групи. Не існує жодного відомого рішення з використання української мови.

Предметом дослідження є діалогова підсистема інтелектуального агента.

Метою роботи є підвищення ефективності діалогової системи інтелектуального агента для взаємодії з користувачем на природній мові за рахунок використання сучасних моделей машинного навчання та їх реалізації на українській мові. Що допоможе підвищити якість роботи прикладних програм по обробці природної

мови при використанні розробленої системи і надасть можливість подальшого розвитку інтелектуальних агентів, які базуються на українській мові.

Головними критеріями ефективності є:

1. Точність та якість отриманих відповідей.
2. Швидкість взаємодії.
3. Зручність використання системи

Для досягнення мети необхідно вирішити наступні задачі:

1. Проаналізувати проблематику теми, що досліджується.
2. Проаналізувати існуючі рішення.
3. Проаналізувати існуючі підходи до побудови інтелектуальних агентів.
4. Проаналізувати існуючі алгоритми, які використовуються для вирішення схожих завдань.
5. Сформулювати вимоги до розробки.
6. Визначити засоби та технології, які можуть бути використані.
7. Розробити архітектуру додатку.
8. Впровадити алгоритм роботи.
9. Знайти або розробити набір даних, який буде використаний для навчання моделі.
10. Реалізувати алгоритмічне і програмне забезпечення.
11. Обробити набір даних.
12. Провести навчання.
13. Провести експериментальне дослідження.
14. Проаналізувати отримані результати.

1.2 Аналіз особливостей

Технології автоматизації переймають усі сфери нашого життя, будь то розвиток розумних міст, розумних будинків, автоматизованих робочих просторів або таких технологій, як смартфони та цифрові персональні асистенти. З кожним новим

розвитком ми просуваємось на крок до більш пов'язаного та цифрового майбутнього. Експерти галузі одностайні у своїй думці, що технологія чат-ботів ще знаходиться в зародковому стані. Ми лише чуємо поверхню того, як може виглядати майбутнє з підтримкою чату.

Одне досить зрозуміло, що чат-боти тут залишаються, і їх розвиток вплине як на бізнес, так і на споживачів. Нинішня реалізація чатів у галузі обслуговування клієнтів пропонує двері до розуміння майбутнього використання чат-ботів для різних аспектів бізнес-операцій. Більше того, лише шляхом випробування ми можемо виявити інноваційні шляхи подальшої реалізації цієї технології.

Якщо ви власник малого або середнього бізнесу, який хоче розширити свій бізнес в Інтернеті, чати можуть зіграти важливу роль у вашій історії успіху. Чат-боти можуть поставити дві найважливіші вимоги до будь-якого онлайн-бізнесу сьогодні - якість продукції та наявність на ринку. Завдяки бізнес-рішенню chatbot, ви можете використовувати розумні технології AI для того, щоб ваші клієнти займалися значущим бізнесом і діловими розмовами. Це розширить вашу базу знань та допоможе вам розробити кращі продукти для своїх клієнтів. Крім того, ви зможете запропонувати клієнтам швидшу та точнішу службу підтримки, підвищивши репутацію вашого бренду. І останнє, і найголовніше, що ви, як ранній користувач, отримаєте більше користі від майбутніх досягнень технології чат-ботів.

У ході дослідження було виявлено ряд особливостей, які роблять чат-ботів такими актуальними, а їх використання неминучим:

1. Доступний у будь-який час.

Я впевнений, що більшість із вас завжди втомлюються, поки оператори з'єднують вас із службою підтримки клієнтів. В середньому люди витрачають близько 7 хвилин, поки їх не призначать оператору. І все ж, більшість користувачів зараз користуються саме цим методом зв'язку з працівниками підтримки, так як вони замінюють інші форми більш повільних методів контактів, такі як електронні листи та телефонні дзвінки.

Оскільки чат-боти – це в основному віртуальні роботи, вони ніколи не втомлюються. Вони будуть працювати щодня протягом року, не вимагаючи перерви. Це покращує ваші клієнтські послуги UX та допомагає вам досягти високих позицій у вашому секторі. Ще одна перевага цієї миттєвої відповіді полягає в тому, що ви також можете вміло скласти свій чат, щоб підтримувати свій імідж та бренд.

2. Потужність обробки.

На відміну від людей, які можуть спілкуватися лише з однією людиною одночасно, чати-боти можуть одночасно спілкуватися з тисячами людей. Незалежно від того, в який час доби або скільки людей звертаються до вас, кожен із них відповість негайно.

Уявіть, що у вас є ресторан, і ви маєте хорошу репутацію в їжі, з якої більшість ваших доходів надходить від доставки. Оскільки попит постійно зростає, у вас буде більше клієнтів, які можуть робити замовлення, але дуже мало персоналу, щоб опрацювати їх усі. Наявність чату дозволить усунути таку проблему та задовольнити кожну людину, забезпечить, щоб жодне замовлення не було пропущено. Такі компанії, як Taco Bell і Dominos, вже використовують чати для організації доставки посилок.

3. Гнучкість.

Чат-боти мають ту перевагу легкого використовувати в будь-якій галузі. На відміну від інших продуктів, де вам доведеться зробити багато розробок і тестування, щоб змінити платформи, чат-боти порівняно легко переміщуються. Треба просто тренувати бота, надаючи правильну структуру розмови та потік, щоб переключити своє поточне поле чи галузь.

Або якщо між двома розділами галузі є багато спільного, можна говорити про підтримку клієнтів та продаж на загальному рівні.

4. Задоволеність клієнтів.

Поведінка людей так чи інакше змінюється від настрою, що не завжди добре для бізнесу. З іншого боку, чат-боти пов'язані з деякими правилами та підкоряються їм, доки вони запрограмовані. Вони завжди будуть ставитись до клієнта ідеально, незалежно від того, наскільки груба людина або наскільки нецензурна мова використовується.

Не кожен замовляє одну і ту ж їжу щодня, вибір людей може змінюватися. У цьому випадку він може використовувати вашу історію замовлень, щоб вносити пропозиції щодо наступного замовлення, дізнаватися інформацію про вашу адресу та багато іншого. Клієнти люблять цю плавну взаємодію і хочуть, щоб усі їх транзакції були максимально простими.

5. Ефективність.

Наймання людини на роботу ніколи не є дешевою справою, і це буде дорого, якщо ваш дохід не високий або цілі продажу не будуть досягнуті і створили б хаос в бізнесі. Через межі людських істот, одна людина може одночасно впоратися лише з однією або двома людьми.

Чат-боти можуть допомогти вирішити цю вікову проблему. Оскільки один Chatbot дорівнює навантаженню працівників, він може легко спілкуватися з тисячами клієнтів одночасно. Нам знадобиться лише декілька людей, щоб приєднатися до розмови, коли це необхідно. Отже, це б різко знизило витрати та призвело б до різкого зростання доходів та задоволеності клієнтів.

6. Швидкість навчання.

Перш ніж ви хочете виконати завдання, спочатку ви повинні навчитися працювати над завданням і тільки тоді виконати його. Існує безперервне навчання, що бере участь у всіх рівнях ієрархії, через які працівник пройде. Також буде багато змін у працівниках, дехто залишиться, когось звільнять, ще когось приєднують тощо.

Що я хочу сказати, працівники змінюються і це факт. І навчання нового працівника займає багато часу, ресурсів та фінансів. Коли з використанням чат-бота цю процедуру можна зменшити практично до нуля.

7. Автоматизація роботи.

Люди, як правило, менш продуктивні, коли отримують повторювану роботу. Нам, людям, зазвичай нудно робити те саме і знову і знову. Тепер чати можуть автоматизувати завдання, які потрібно виконувати часто і в потрібний час. І зараз вже є численні боти, які автоматизують повторювані завдання. Це допомагає людям економити час та бути більш продуктивними.

8. Альтернативний канал збуту.

Чат-боти можуть продавати вашу продукцію для вас, оскільки вони в режимі онлайн 24/7. Недавнє дослідження показало, що 70% людей вважають за краще текстові повідомлення, а не викликати. І тепер, коли домінує Amazon в електронній комерції, нікому не потрібно їздити в магазин, щоб придбати речі. Як результат, люди прагнуть виходити в Інтернет для задоволення своїх потреб. Це дає можливість продавати товари, які б відповідали потребам кожного замовника. Крім того, чат-боти запам'ятають відповіді замовника та адаптуються до цього. Роблячи це, вони створюють особистий рівень обслуговування, який тісно відображає взаємодію людини.

Бренди електронної комерції, такі як H&M, eBay shopbot, Tommy Hilfiger зараз продають багато товарів за допомогою ботів. Amazon також робить великі ставки з набагато просунутішим розпізнаванням голосу AI під назвою "Alexa", яке набирає популярності в наш час.

9. Персональний асистент.

Люди можуть використовувати ботів як особистий радник моди для рекомендацій щодо одягу, або запитати поради щодо торгівлі у фінансового бота, або щодо місця для відвідування у подорожнього бота тощо. Це допоможе користувачам отримати більш особистий контакт з чат-ботом. Крім того, чат-бот

запам'ятає всі ваші варіанти та надасть вам відповідні варіанти наступного разу, коли ви його відвідаєте. Помітні приклади – Trīm, особистий бот з фінансів; Тейлор – помічник подорожей, бот CNN для персоналізованих новин.

1.3 Огляд існуючих рішень

Вартість та потужність чат-бота залежать від технології, яку він використовує. Виділяють дві категорії:

- Чат-боти на основі команд;
- Чат-боти на основі AI.

Основа на командах, покладаються на банк даних відповідей та евристики. Боти відповідають, вибираючи відповідь, що відповідає контексту запиту. Чат-боти на основі команд не можуть створювати нові тексти.

Характеристика командних чатів:

- Ці боти можуть відповісти на обмежений набір питань. Боту буде потрібна допомога вручну для нетипових питань;
- Чат-бот не може виконувати функції поза своїм кодом;
- Щоб відповісти на запитання, чат повинен спочатку зрозуміти це. Чат-боти на основі команд використовують пошук шаблонів або динамічний пошук для розуміння та відповіді на запитання.

Чат-боти на основі AI або ML можуть відповісти на неоднозначні запитання. Це означає, що вам не потрібно бути конкретними, коли задаєте питання цим чатовим роботам. Чат-боти створюють відповіді з нуля, використовуючи природну обробку мови. Ці чати з часом стають розумнішими, навчаючись на минулих питаннях та відповідях.

Обидві категорії мають свої переваги та недоліки. Агенти на основі команд можуть дати більш надійні та граматично правильні відповіді, але не можуть відповісти на запитання, що не знаходяться в їх базі знань. З іншого боку, для

машинного навчання чатовим роботам потрібна величезна кількість навчальних даних і вони можуть затруднятися дати більш швидкі та точні відповіді.

1.3.1 Огляд інтелектуальних помічників на основі команд

1.3.1.1 Mondly

Mondly – це платформа для вивчення мови, яка фокусується на інтерактивних онлайн-уроках. Інтелектуальний агент використовує технологію розпізнавання мовлення, але ви також можете дотримуватися письма. Чат-бот закликає користувачів практикувати свою англійську, іспанську, німецьку чи французьку мови через щоденні рольові сценарії, такі як бронювання готелів або відвідування ресторану.



Рисунок 1.1. Приклад роботи montly chatbot

Окрім звичайного розмовного чату, Mondly випустив програму Virtual Reality для Oculus, яка забезпечує тривимірне середовище для підвищення рівня залучення користувачів. Вони говорять, що у наш час користувачі повинні навчатися на як можна більш реалістичних прикладах.



Рисунок 1.2. Приклад віртуального середовища для montly chatbot

1.3.1.2 Buoy Health Chatbot

Buoy Health Chatbot – приклад інструменту ШІ, який повинен імітувати розмову з лікарем. Замість використання простих дерев рішень Buoy chatbot використовує свою базу даних з десятків тисяч клінічних записів, щоб динамічно вибирати найкращі питання інтерв'ю пацієнта.

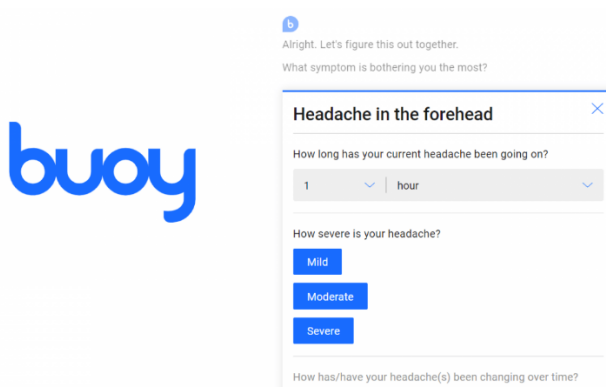


Рисунок 1.3. Приклад роботи buoy health chatbot

Агент був протестований на великій групі пацієнтів, яких опитував чат-бот і яких потім оглядав лікар. Професійний діагноз відповідає дійсності більш ніж у 90% випадків. У разі будь-яких тривожних симптомів додаток негайно пропонує порадитись з реальним лікарем.

1.3.1.3 Realbot Chatbot

Компанія спеціалізується на створенні чат-ботів, які можуть допомогти агентам з нерухомості. Агент з нерухомості оснащений спеціалізованими потоками розмов, такими як *«Що є в мене вдома?»* або *«Орендарі та управління майном»*. Вони можуть бути досить корисними для агентств нерухомості для фільтрації найбільш перспективних потенційних клієнтів.

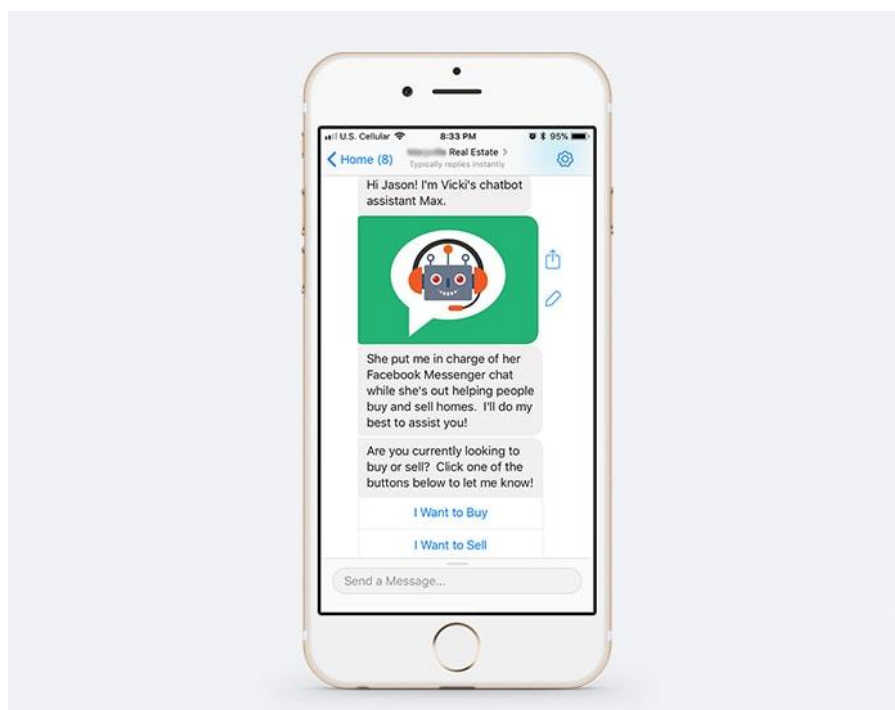


Рисунок 1.4. Приклад роботи realbot chatbot

Сила Realty Chatbot полягає в розробці розмов, розроблених спеціально для галузі нерухомості, і типових сценаріях, з якими агенти мають справу щодня. Намагаючись покращити обслуговування клієнтів, хороші шаблони чатів – це половина успіху.

1.3.2 Огляд інтелектуальних помічників на основі нейронних мереж

1.3.2.1 Casper

Якщо ви страждаєте від безсоння, ви знаєте це почуття майже задушливої самотності – ідея про те, що всі інші люди в світі спокійно відпочивають, а ваш власний розум зраджує вас турботами та сумнівами – є однією з найгірших частин не в змозі спати.

Каспер, також відомий як Insomnobot 3000 (що справді є одним із найбільш чітких, ретро-футуристичних імен для чат-бота, з яким я коли-небудь стикався), розмовний агент, який має на меті дати можливість з кимось поговорити, коли решта світу відпочиває.

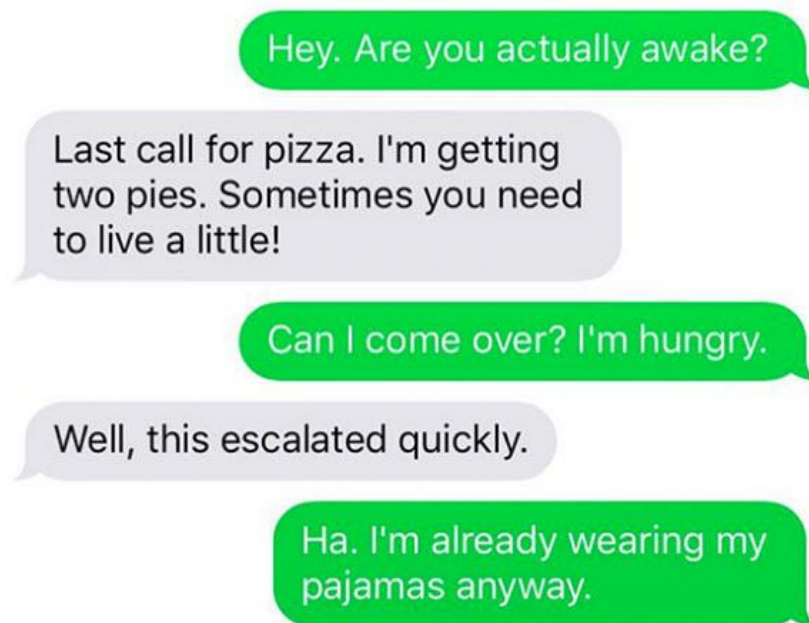


Рисунок 1.5. Приклад роботи insomnobot 3000

На даний момент Insomnobot 3000 трохи недопряцьований. Як ви можете бачити на знімку екрана, відповіді, запропоновані агентом, не зовсім правильні, але бот все одно підкреслює, як розмовні агенти можуть бути використані.

1.3.2.2 Roof AI

Якщо ви працюєте в маркетингу, ви, напевно, вже знаєте, наскільки важливим є завдання надання послуги потрібний час. Особливо якщо ви працюєте з великою клієнтською базою.

Увійдіть у Roof AI, чат-робот, який допомагає маркетологам нерухомості автоматизувати взаємодію з потенційними клієнтами та контакту з клієнтами через соціальні мережі. Бот визначає потенційних клієнтів через Facebook, потім реагує майже миттєво дружньою, корисною та розмовною тональністю, що дуже нагадує реальну людину. Ґрунтуючись на інформації користувача, Roof AI пропонує потенційним клієнтам надати трохи більше інформації, перш ніж автоматично призначити потенційну інформацію агенту з продажу.

Hi, I am looking for a one bedroom apartment in Montreal. My budget is \$500k. I need an indoor parking.

Hi! I'm Roof the real estate bot. When are you looking to move?

I want to move by the end of February!

Awesome! One of my colleagues will be with you shortly 😊

Рисунок 1.6. Приклад роботи roof AI

Однією з ключових переваг Roof AI є те, що він дозволяє агентам з нерухомості відповідати на запити користувачів негайно, незалежно від того, чи є представник служби обслуговування клієнтів або агент з продажу. Це може мати

істотний вплив на коефіцієнти конверсії. Це також виключає потенційні втрати через недостатнє реагування.

Загалом, Roof AI – це надзвичайно точний бот, який багато ріелторів, ймовірно, вважають незамінним. Бот ще розробляється, хоча зацікавлені користувачі можуть зарезервувати доступ до Roof AI через веб-сайт компанії.

1.3.2.3 Mya Chatbot

Mya – приклад помічника з підбору персоналу AI chatbot. Він використовує NLP та машинне навчання для автоматизації процесу відбору. L'Oréal використовує Mya для скорочення завантаженості відділу кадрів, який щороку отримує понад мільйон заявок на роботу.

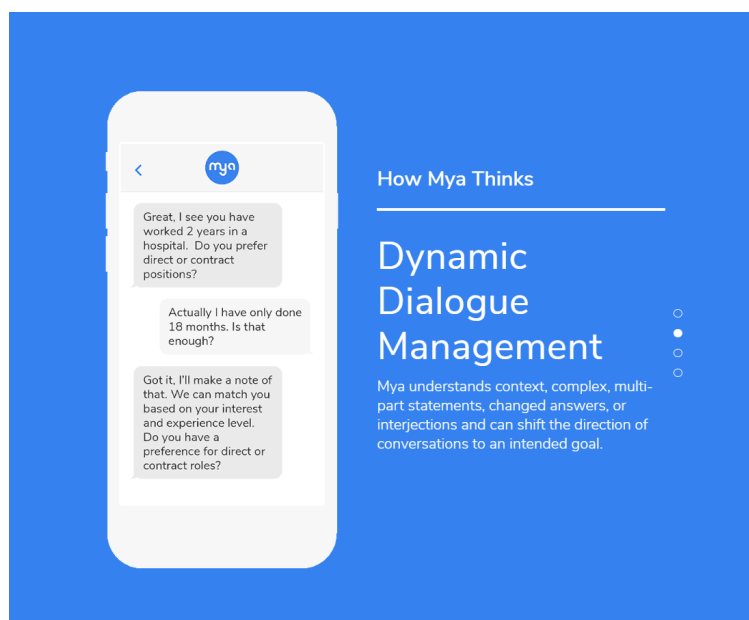


Рисунок 1.7. Приклад роботи mya chatbot

Mya дає точні та залежні від контексту відповіді. Чат-бот усуває проблеми найму та вибирає найперспективніших кандидатів, до яких пізніше можуть звернутися фахівці з персоналу, щоб завершити процес прийому на роботу. Mya робить 75% роботи.

1.3.3 Огляд особливих реалізацій інтелектуальних помічників

1.3.3.1 Visual Dialog

Visual Dialog – це візуальний чат-бот, який може інтерпретувати зображення. Чат-бот базується на комп'ютерному зорі та нейронних мережових технологіях. Користувачі можуть завантажувати зображення безпосередньо через вікно чату. Visual Dialog надішле повідомлення з описом того, що на малюнку.

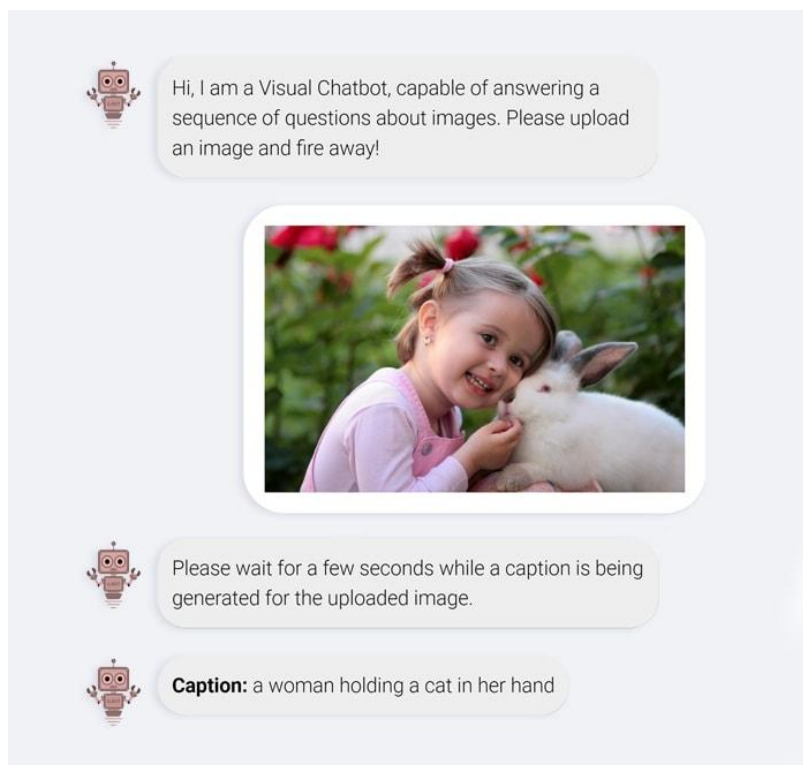


Рисунок 1.8. Приклад роботи visual dialog

Гра з візуальним чат-ботом може бути дуже захоплюючою і цікавою. Особливості розпізнавання зображень іноді використовуються і в чатах електронної комерції. Візуальні чати від популярних брендів, таких як Nike, можуть використовувати фотографії, зроблені вашим смартфоном, щоб знайти продукти або запропонувати варіанти налаштування.

1.3.3.2 Hipmunk

Hipmunk – один з найбільш інноваційних прикладів чат-ботів. По-перше, Hipmunk допоможе нам з ідеями подорожей. Визначившись, куди і коли ми хочемо вилетіти, він перегляне мережу в пошуках найкращих рейсів. Він також перевірить, який готель ми можемо орендувати в цей час, і розповість про витрати на оренду. Пізніше він перенаправляється на веб-сайт із готелем, де ми можемо побачити ціни, фотографії та думки інших користувачів!

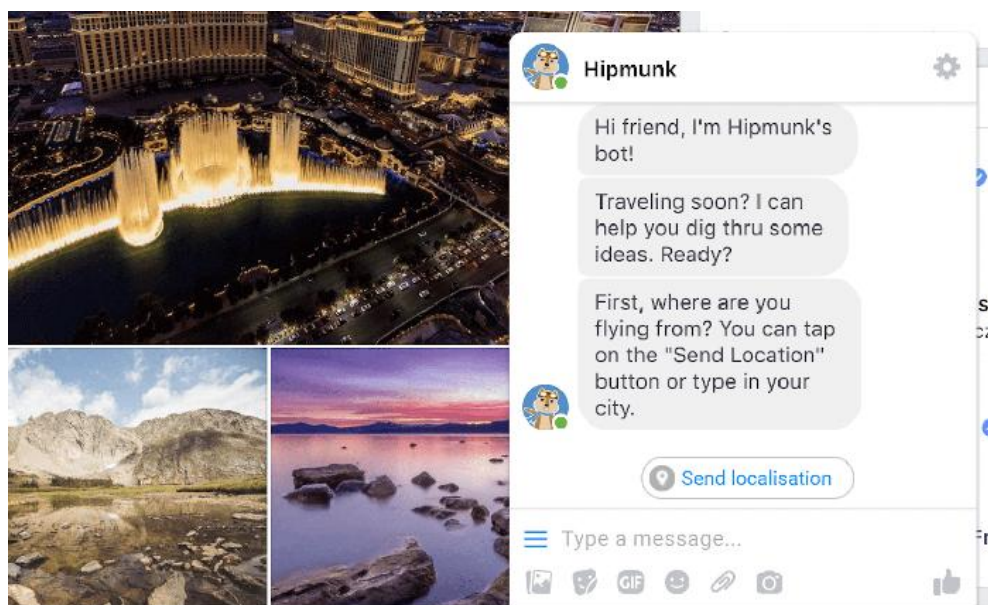


Рисунок 1.9. Приклад роботи hipmunk

Hipmunk придумав ідею, як заощадити час своїх клієнтів за допомогою розмовного інтерфейсу, і завдяки цьому вони приваблюють ще більше людей, зацікавлених подорожувати, використовуючи їх послуги.

1.3.3.3 UNICEF

Міжнародна некомерційна адвокатська організація UNICEF використовує чат-ботів, щоб допомогти людям, що живуть у країнах, що розвиваються, висловитись про найактуальніші потреби у їхніх громадах.

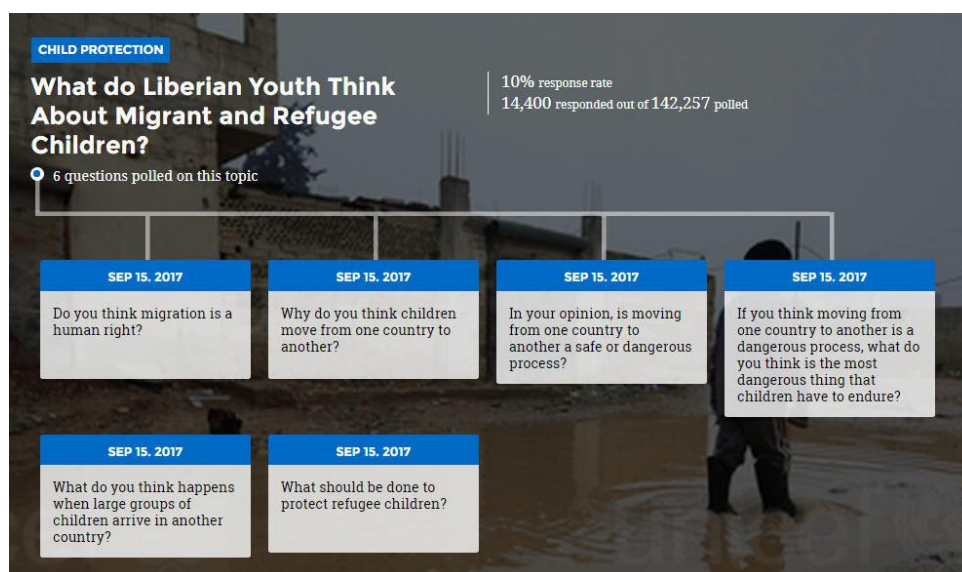


Рисунок 1.10. UNICEF

Бот, званий U-Report, зосереджується на масштабному збиранні даних за допомогою опитувань – це не бот для балакучих. U-Report регулярно надсилає підготовлені опитування щодо низки нагальних соціальних питань, і користувачі (відомі як «U-Reporters») можуть відповісти своїм внеском. Потім ЮНІСЕФ використовує цей відгук як основу для потенційних рекомендацій щодо політики.

1.4 Аналіз підходів до побудови інтелектуальних агентів

На сьогодні відомий ряд підходів, які використовуються при створенні інтелектуальних агентів. Перед початком пошуку кожне слово фрази потрібно перевести в нейтральний стан. Для цього використовуються такі методи як стемінг, лематизації, пошук схожих слів за словником, тощо. Коротко розглянемо самі методи.

1. Кожне слово приводиться до вектору Word2Vec або Glove. Підсумувати всі слова-вектори фрази.

2. Використовувати нейромережі, класифікуючи фрази в номер команди. Для цього створюється невеличкий датасет «фраза» – «номер команди». Після чого використовуються LSTM, GRU нейрони або одновірні згортки Conv1d.

3. Використовувати складні регулярні вирази. При цьому агент працює за схемою «команда» – «регулярний вираз». Що у свою чергу дає різні закінчення та різні слова.

4. Пошук сутностей у фразі, а далі уточнювати. Можете бути використано при наданні не точної інформації, або ж при роботі з широким колом задач.

5. Пошук частин мови, відштовхуючись від іменників і дієслів. При реалізації цього методу необхідно реалізовувати дерево.

6. Пошук по найбільшому перетину множин слів.

Розглянемо приклад реалізації для першого методу. Допустимо ми маємо список базових команд:

- Відкрий браузер;
- Включи музику;
- Замов їжу;
- Викликай таксі;
- Напиши текст;
- Відправ електронного листа;
- Постав будильник;
- Запиши нагадування;
- Зміни мову;
- Закрий вкладку.

Спочатку потрібно провести лематизацію кожної фрази і видалення стоп-слів

```
preprocess_text('сделай мне чего-то покушать')
```

```
'сделать что-то покушать'
```

```
preprocess_text('я опаздываю, закажи такси')
```

```
'опаздывать заказывать такси'
```

Рисунок 1.11. Приклад лематизації

Далі кожне слово перетвореної фрази відображаємо в вектор і підсумовуємо вектор слів в вектор фрази.

```
get_vec('я опаздываю, закажи такси')|
array([-1.7968918e+00,  6.4863890e-01, -4.3391423e+00,  2.5097926e+00,
       -1.3486636e-01, -2.0945344e+00,  1.2578716e+00,  2.4579520e+00,
       -2.4501736e+00,  2.3522878e-01,  1.6352599e+00, -2.8476582e+00,
        9.9342728e-01,  3.5668712e+00,  1.2617840e+00,  4.8110332e+00,
        1.8528476e+00, -2.6652157e+00, -1.6782639e+00, -1.8418138e+00,
        1.1326343e-02,  3.8728304e+00,  1.2627807e+00, -2.4413035e+00,
        5.2752674e-02, -8.0001249e+00,  4.7649922e+00, -1.0555379e+00,
        1.5655214e+00, -3.5883381e+00, -1.4905419e+00,  1.6576748e+00,
       -1.8687749e-01,  2.3763909e+00,  3.7991807e+00,  4.8204527e+00,
        3.3429594e+00,  7.2259021e-01, -4.1156149e+00,  2.3608408e+00,
        1.5862646e+00, -1.4994133e+00, -9.0723920e-01,  2.2480853e+00,
        3.8099661e+00,  3.1441519e+00,  2.2273872e+00, -1.0148538e+00,
       -2.1033585e-01,  2.5309687e+00,  1.2529368e+00, -2.2407670e+00,
       -2.9847946e+00, -3.8947296e+00,  6.0097651e+00,  3.7903304e+00,
        1.1803641e+00, -5.7860823e+00, -2.2043579e+00,  2.9816658e+00,
       -1.2652884e+00,  2.5090082e+00, -9.3071520e-01,  2.9038310e+00,
       -6.5632396e+00, -4.3539486e+00, -4.7275484e-01, -3.7017703e+00,
       -4.5858479e+00, -2.2164583e+00,  8.1658649e-01,  2.9526889e+00,
        5.0176489e-01, -1.7398959e+00, -4.8825836e-01, -2.1333432e+00,
       -3.7610903e+00,  2.4662390e+00, -4.1699845e-01, -3.8535483e+00,
       -6.7018509e-01, -4.3530865e+00,  8.0320778e+00,  7.2134476e+00,
       -3.9380138e+00,  2.1138892e+00,  5.2496710e+00,  3.3246150e+00,
```

Рисунок 1.12. Приклад формування вектору фрази

Таким чином кожен базову команду перетворюємо в свій вектор, з якими потім будемо порівнювати призначені для користувача фрази. Для порівняння використовується косинус кута між векторами. Відповідно, чим менше кут між векторами, тим більше вони схожі, тим більше значення косинуса.

```
nearest('дай покушать')
{'Открой браузер': 0.11725078,
 'Включи музыку': 0.070676334,
 'Закажи еду': 0.24351554,
 'Вызови такси': -0.013970826,
 'Напиши текст': 0.04669959,
 'Отправь электронное письмо': 0.051718593,
 'Поставь будильник': 0.08935451,
 'Запиши напоминание': 0.14313854,
 'Смени язык': 0.0042991014,
 'Закрой вкладку': -0.015158206}
```

Рисунок 1.13. Приклад зіставлення вектору фрази та базового набору команд
Для фрази «дай поїсти» найбільш близьким є вектор «замов їжу», що, в принципі, логічно. Однак, модель дуже чутлива до слів в ідеальній фразі.

Переваги моделі: дозволяє обробляти абсолютно будь-які фрази, ми обмежені тільки словником Word2Vec, також в більшості випадків модель дає хороші результати по визначенню базових команд.

Недоліки: дуже чутлива до «ідеальних» словами в базових командах.

1.5 Постановка задачі

У результаті проведення аналізу існуючих рішень, предметної області, формування об'єкту дослідження та предмету дослідження, а також аналізу підходів до побудови інтелектуальних агентів, був сформований висновок, що створений продукт повинен відповідати наступним вимогам:

- Вся взаємодія з користувачем проходить виключно, з використанням української мови;
- Спробувати досягти точності розпізнавання вільної мови у районі 90%;
- У результаті соціального тестування, відсоток задоволених зручністю користувацького інтерфейсу повинен бути близько 75%;
- Розроблене програмне забезпечення повинно мати можливість інтеграції у більшість популярних платформ;
- Простота у використанні;
- Інтуїтивно зрозумілий інтерфейс.

Висновки до розділу

У першому розділі були визначені область використання, об'єкт дослідження, предмет дослідження. Після чого був здійснений аналіз існуючих рішень для різних видів інтелектуальних агентів, та аналіз підходів і технологій, які використовуються при реалізації. На основі чого були визначені недоліки і переваги цих рішень. В результаті проведеного аналізу сформульована постановка задачі, наведене призначення, а також зазначення вимоги до розробки.

РОЗДІЛ 2. ІНФОРМАЦІЙНА МОДЕЛЬ

2.1 Функціональна модель

Діалогові агенти можуть проводити людиноподібну взаємодію з людиною за допомогою обробки природних мов (NLP). Вони є розробленими версіями чат-ботів, які можуть обслуговувати автоматизовані відповіді або відповідь замовчуванням. Діалогові агенти – це по суті розумні чат-боти.

Основний розмовний інтелектуальний інтерфейс приймає запит від користувача через текст або голос і передає його наступним модулям:

Модуль підключення: це розмовна платформа, інтерфейс і може бути веб-сайтом або додатком, де користувач взаємодіє з AI.

Модуль введення: запит користувача подається до цього модуля, для розуміння природної мови.

Діалогове управління: цей модуль містить текст NLU та підтримує контекст розмови. Потім використовується модель машинного навчання, таку як RNN та LSTM, щоб запропонувати відповідні дії. Цей модуль також можна підключити до коду управління програми або бази даних для виконання додаткової обробки на основі запропонованих дій.

У наведеному нижче прикладі він визнає, що користувач запитує погоду та підключається до API погоди, щоб виконати відповідну дію.

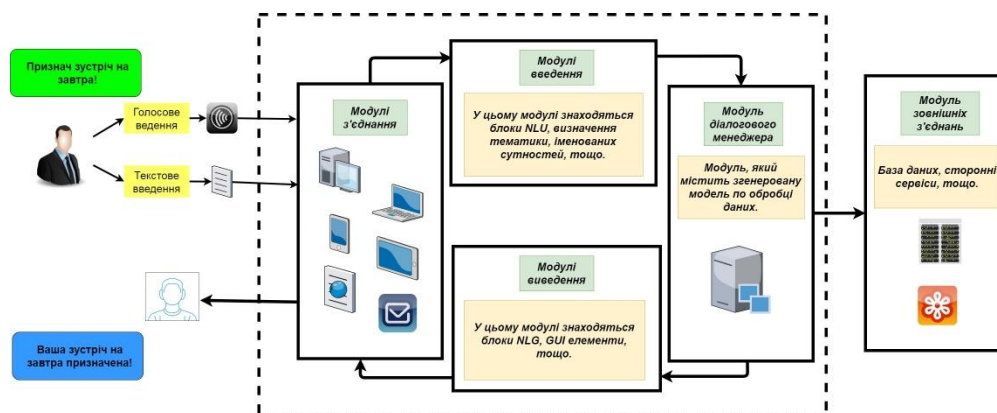


Рисунок 2.1. Функціональна схема додатку

Вихідний модуль: цей модуль виконує створення природної мови, або вибір з даних, які використовуються, як відповідь, щоб повернути користувачеві відповідну відповідь.

Користувач надсилає назад відповідь AI за допомогою тексту або голосу через модуль з'єднувача .

У інтелектуального агента є два наступні компоненти:

1. NLU , який інтерпретує повідомлення користувача.
2. Core , який відстежує розмову і вирішує, що буде далі. Core пропонує найбільш прийнятні дії.

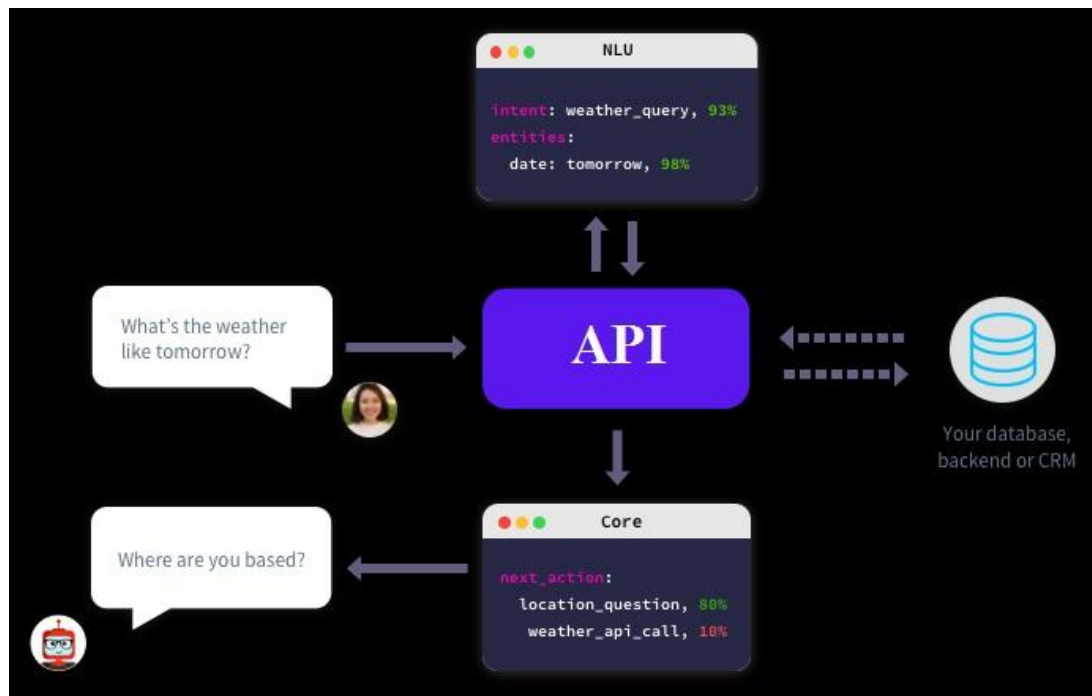


Рисунок 2.2. Компоненти системи

2.1.1 NLU

NLU обробляє введений користувачем текст і розуміє, що користувач намагається сказати. В основному, він приймає текст користувача як вхідний і витягує з нього намір та сутності .

Намір: відображає мету введення користувача, що він хоче зробити. Текст введення користувача спочатку векторизується, а потім з тексту витягуються сутності, які потрібні для здійснення певних дій.

Сутність: суб'єкт представляє термін або об'єкт, що відповідає вашим намірам і що забезпечує конкретний контекст для наміру.

Для прикладу при введенні: "Яка погода завтра?", — ми отримаємо:

Намір: `"request_weather"` – Користувач запитує про погоду.

Сутності: `{"дата": "завтра"}` – На котру дату користувач запитує погоду?

Ми можемо визначити додаткові наміри, якщо я вони потрібні для реалізації певного функціоналу. Наприклад, ви також можете визначити об'єкт "місце", щоб ви могли зрозуміти, з якого місця слід дізнатися погоду.

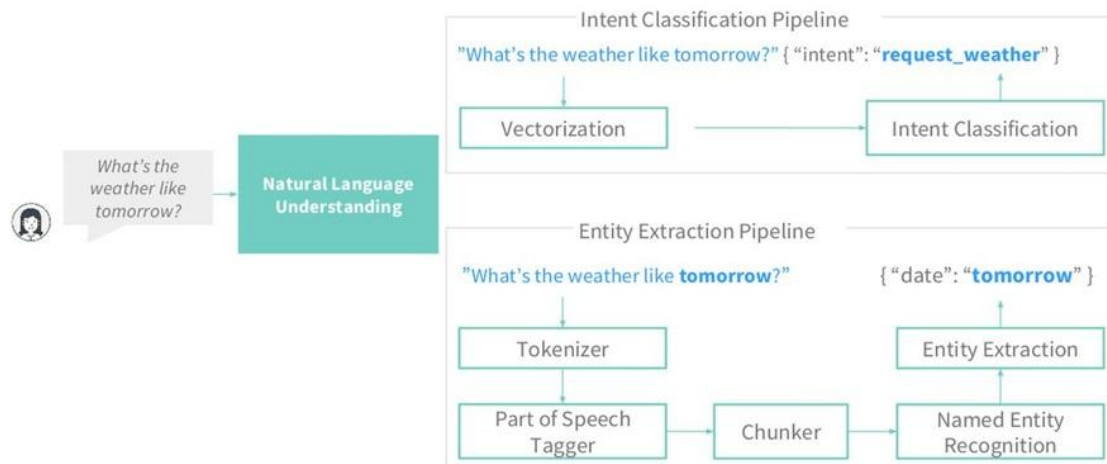


Рисунок 2.3. Компонент системи NLU

2.1.2. Core

Після вилучення даних з користувацького введення, наміри та сутності передаються до Core частини. Core вирішує, що буде далі в цій розмові. Він використовує управління діалогом на основі машинного навчання, щоб передбачити наступні найкращі дії на основі інформації від NLU, історії бесіди та ваших даних при навчанні.

Скажімо, ваш користувач запитує місце розташування ресторану, не вказуючи його ім'я.

1. Ваша система отримає намір "*looking_for_restaurant*" і розпочне нову мету "*find_restaurant*".
2. Буде помічено, що для досягнення цієї мети потрібно знати назву ресторану. Тому він запитає у користувача назву.
3. Коли користувач відповість, він спочатку проаналізує цю відповідь, щоб побачити, чи містить вона назву ресторану. Якщо знайдено, відбудеться збереження ім'я в його контексті.
4. Нарешті система побачить, чи може тепер закінчити ціль «*find_restaurant*». Оскільки назва ресторану тепер відома, він може знайти місцезнаходження ресторану та повідомити його користувачеві.

Цей тип управління діалогом працює на основі поведінки, а не стану. Простіше керувати різними способами задати одне і те ж питання, переключення контексту або прийняття рішень на основі того, що ви знаєте про користувача.

Маючи доступні масивні дані, інтуїтивно можна побудувати розмовну систему на основі пошуку, оскільки методи пошуку інформації швидко розвиваються. Враховуючи висловлювання користувача як запит, система шукає відповіді кандидатів шляхом відповідності показників. Ядро розмовних систем, заснованих на пошуку, формулюється як відповідність між висловлюванням запиту та відповідями кандидатів. Типовим способом узгодження є вимірювання внутрішнього продукту двох, що представляють особливості векторів для запитів та відповідей кандидатів у трансформованому просторі Гільберта. Намагання з моделювання зводиться до пошуку відображення від початкових входів до функціональних векторів, що називається навчанням представлення. Існує двоступінчаста техніка пошуку для пошуку відповідних відповідей з масивного сховища даних. Процес пошуку складається з швидкого ранжирування за стандартом TF-IDF вимірювання та процес переосмислення з використанням

функцій, орієнтованих на розмову, розроблених з досвіду людини. Системи для вибору найбільш підходящої відповіді на запит із пар запитання-відповіді, використовуючи статистичну модель мови в якості багатомовної пошуку інформації. Більшість сильних пошукових систем вивчають уявлення з глибокими нейронними мережами (DNN). DNN – це високо автоматизовані навчальні машини; вони можуть витягувати основні абстрактні особливості даних автоматично, досліджуючи кілька шарів нелінійного перетворення. Переважаючі DNN для моделювання рівня речень включають в себе нейромережі згортання (C-NN) та періодичні нейронні мережі (RNN). Серія методів узгодження може бути застосована до коротких текстових розмов для систем на основі пошуку.

Існує відома за тематична мережа, яка використовує слова теми, у відповідності повідомлення та відповідь.

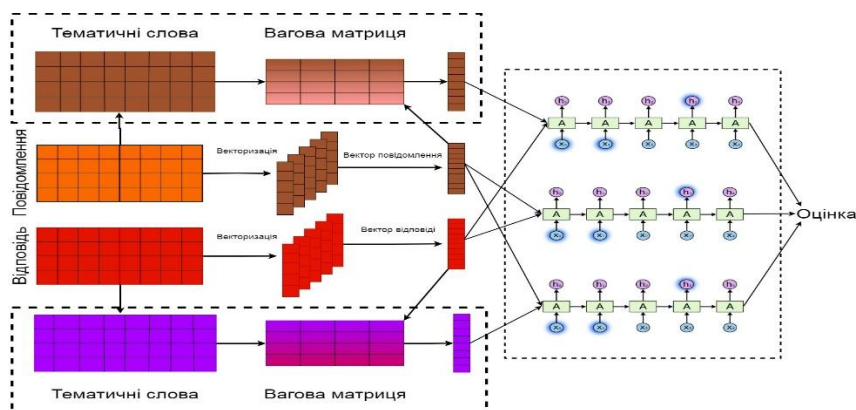


Рисунок 2.5. Нейронна мережа системи

Кодер RNN створює послідовність контекстних лексем по одному і оновлює його прихований стан. Після обробки всієї послідовності контексту він створює остаточний прихований стан, який включає в себе сенс контексту і використовується для генерування відповіді.

Модуль розуміння природних мов, який використовується менеджером діалогів, який обробляє введення користувача для пошуку ключових слів, за допомогою яких можна зрозуміти дії, які слід вжити.

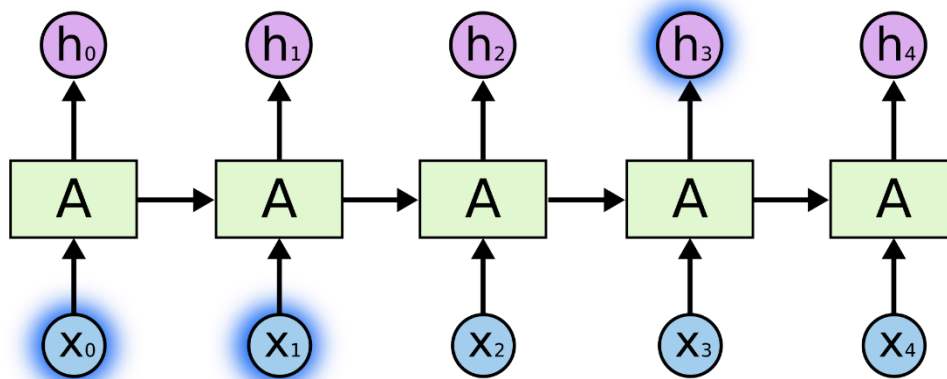


Рисунок 2.6. Рекурентна нейронна мережа

Модуль генерації природної мови, який генерує відповіді з інформації, зібраної менеджером діалогу.

З часом ми зіткнулися з реальною еволюцією в розробці розмовних агентів, орієнтованих на завдання, через наявність глибоких методів навчання.

Дана картинка нижче ілюструє процес аналізу настроїв у створеному користувачем контексті.

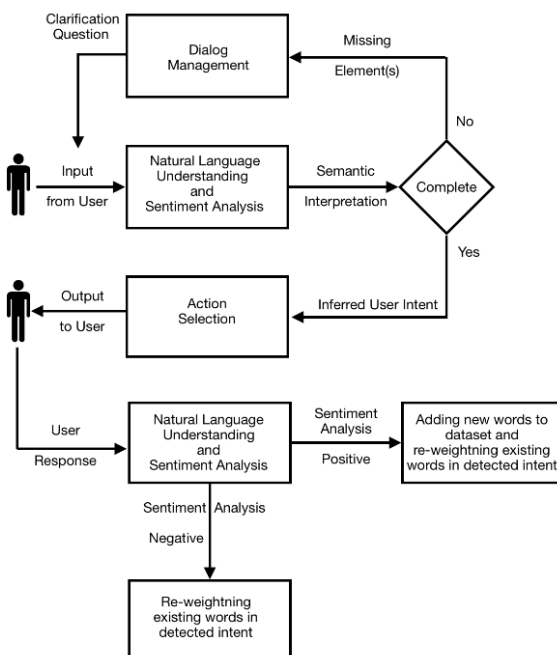


Рисунок 2.7. Схема аналізу настрою повідомлення

2.2 Підготовка даних

2.2.1 ВЕСУМ

У результаті пошуків, як корпус для навчання був обраний набір даних, над яким вже працювали з даною метою, а саме, для формування текстового комплексу для можливості його використання машинному навчанні.

Для побудови лематизованої версії було використано словник ВЕСУМ. Це — великий електронний словник української мови. Він містить слова та моделі з відповідними тегами, серед яких можна знайти таку інформацію як:

- Спеціальні теги для слів: *slang*, *positive*, *rare*, *negative*, тощо;
- Слова-замінники для сленгових висказувань;
- Створена відповідність між стандартною формою слова та його відмінками;
- Управління відмінками для прикметників.

Даний проект дає можливість згенерувати всі форми для базового слова за правилами, які прописані у спеціальному конфігураційному файлі.

Для використання інструменту є ряд технічних обмежень:

- Java (JDK ≥ 8);
- 4Гб вільної пам'яті.

2.2.2 MITIE

На основі цього корпусу була натренована модель, яка дає можливість пошуку NER сутностей. З цією метою використовується бібліотеку MITIE. Цей проект забезпечує безкоштовні (навіть для комерційного використання) сучасні засоби вилучення інформації. Поточний випуск включає інструменти для вилучення названих об'єктів і виявлення бінарних зв'язків, а також інструменти для навчання користувацьких екстракторів та детекторів відношень.

MITIE побудований на версії dlib , високоефективної бібліотеки машинного навчання, MITIE використовує декілька сучасних методик, включаючи distributional word embeddings [2] та Structural Support Vector Machines[3]. Проект пропонує декілька заздалегідь підготовлених моделей, що надають різний рівень підтримки як для англійської, іспанської, так і німецької мов, використовуючи різні мовні ресурси. Основне програмне забезпечення MITIE написано на C ++, але прив'язки для декількох інших мов програмного забезпечення, включаючи Python, Java, C та MATLAB, дозволяють користувачеві швидко інтегрувати MITIE у власні програми.

2.2.3 Dlib

Dlib – це сучасний інструментарій C ++, що містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення на C++ для вирішення реальних проблем. Він використовується як в галузі, так і в наукових колах у широкому діапазоні областей, включаючи робототехніку, вбудовані пристрої, мобільні телефони та великі високопродуктивні обчислювальні середовища. Відкрите джерело ліцензування Dlib дозволяє вам використовувати його в будь-якій програмі безкоштовно. Є ряд особливостей, які варто відзначити.

Документація:

- На відміну від багатьох проектів з відкритим кодом, цей забезпечує повну та точну документацію для кожного класу та функції. Існують також режими налагодження, які перевіряють задокументовані передумови для функцій. Якщо це ввімкнено, він виявить переважну більшість помилок, викликаних неправильним викликом функцій або неправильним використанням об'єктів;
- Наведено багато прикладних програм.

Портативний код високої якості:

- Бібліотека регулярно тестується в системах MS Windows, Linux та Mac OS X. Однак він повинен працювати в будь-якій системі POSIX і використовувався в Solaris, HP-UX та BSD;
- Для використання бібліотеки не потрібні інші пакети. Потрібні лише API, які надаються поза коробкою ОС;
- Перед тим, як використовувати бібліотеку, не потрібен крок установки або налаштування;

Алгоритми машинного навчання

- Глибоке навчання;
- Звичайні вектори для підтримки класифікації та регресії на основі SMO;
- Методи зменшеного рангу для масштабної класифікації та регресії;
- Відповідні векторні машини для класифікації та регресії;
- Інструменти класифікації класичного загального призначення;
- MultiClass SVM;
- Інструмент для вирішення проблеми оптимізації, пов'язаної зі структурними векторами підтримки ;
- Структурні інструменти SVM для маркування послідовностей;
- Структурні інструменти SVM для виявлення об'єктів у зображеннях, а також більш потужні (але повільніші) засоби глибокого навчання для виявлення об'єктів;
- Структурні інструменти SVM для маркування вузлів у графіках;
- Багатошарові перцептрони.

Числові алгоритми

- Для матричного об'єкта визначено численні лінійні алгебраїчні та математичні операції, такі як розкладання значення сингулярного значення, перенесення, триггерні функції тощо;

- Оптимізація без похідних, обмежених коробкою, за допомогою алгоритму BOBYQA;
- Кілька квадратичних програмних рішень;
- Комбінаторні інструменти оптимізації для вирішення оптимальних задач та задач на мінімізацію/максимальний потік, а також алгоритм СКУ для пошуку найбільш ймовірного дерева розбору.

2.2.4 Word2vec

На основі отриманого корпусу будується модель word2vec.

Word embedding – одне з найпопулярніших представлень словникового складу документа. Він здатний фіксувати контекст слова в документі, смислову та синтаксичну схожість, співвідношення з іншими словами тощо. Точніше кажучи, це векторне зображення певного слова.

Word2Vec – одна з найпопулярніших методик вивчення вбудовування слів за допомогою дрібної нейронної мережі. Він був розроблений Томасом Міколовим у 2013 році в Google.

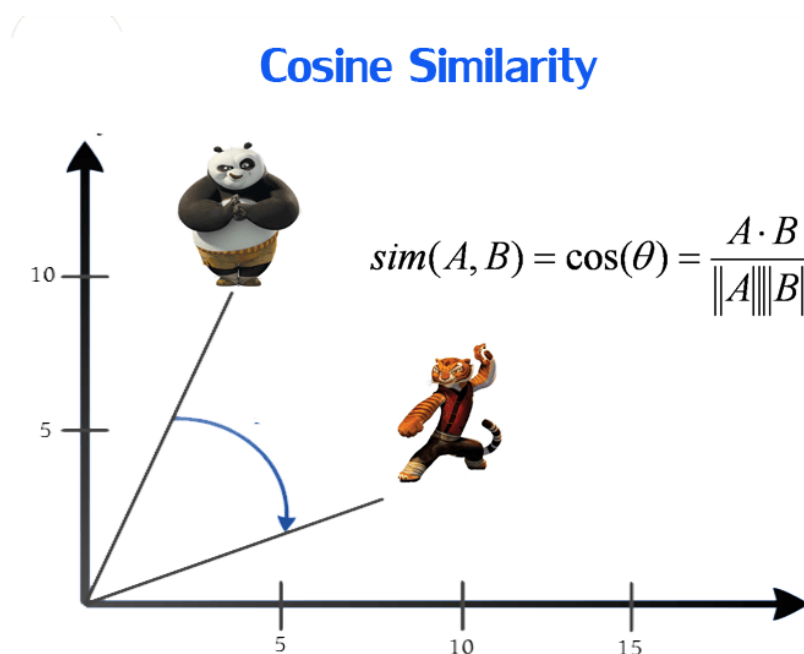


Рисунок 2.8. Word2Vec

Наша мета – щоб слова з подібним контекстом займали близькі просторові позиції. Математично косинус кута між такими векторами повинен бути близький до 1, тобто кут, близький до 0.

Інтуїтивно ми вводимо певну залежність одного слова від інших слів. Слова в контексті цього слова отримали б більшу частку цієї залежності.

Word2Vec – метод побудови такого відображення. Її можна отримати за допомогою двох методів (обидва із залученням нейронних мереж): Skip Gram та Common Bag Of Words (CBOW).

2.2.4.1 CBOW

Цей метод приймає контекст кожного слова як вхідний і намагається передбачити слово, що відповідає контексту. Розглянемо наш приклад: Чудовий день.

Нехай вхід до Нейронної мережі буде словом, чудовий. Зауважте, що тут ми намагаємось передбачити цільове слово (день), використовуючи одне вхідне слово з контексту. У процесі передбачення цільового слова ми вивчаємо векторне подання цільового слова.

Заглянемо глибше у власне архітектуру.

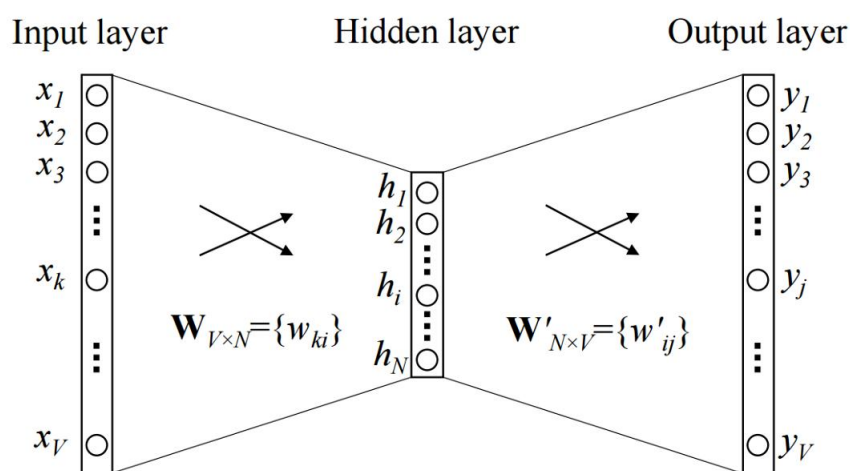


Figure 1: A simple CBOW model with only one word in the context

Рисунок 2.9. CBOW для одного слова

Вхідне слово – це один зашифрований вектор розміром V . Прихований шар містить N нейронів, а вихід – знову V -довжину вектора, причому елементи є значеннями softmax.

Давайте розберемо терміни на малюнку:

1. $W_{V \times N}$ – вагова матриця, яка відображає вхід x у прихований шар ($V * N$ розмірна матриця).
2. $W'_{N \times V}$ – вагова матриця, яка відображає вхід з прихованих шарів на кінцевий вихідний шар.

Нейрони прихованого шару просто копіюють зважену суму входів у наступний шар. Немає активації, як sigmoid, tanh або ReLU. Єдина нелінійність – це обчислення softmax у вихідному шарі.

Але вищевказана модель використовувала єдине контекстне слово для прогнозування цілі. Ми можемо використовувати кілька контекстних слів, щоб зробити те саме.

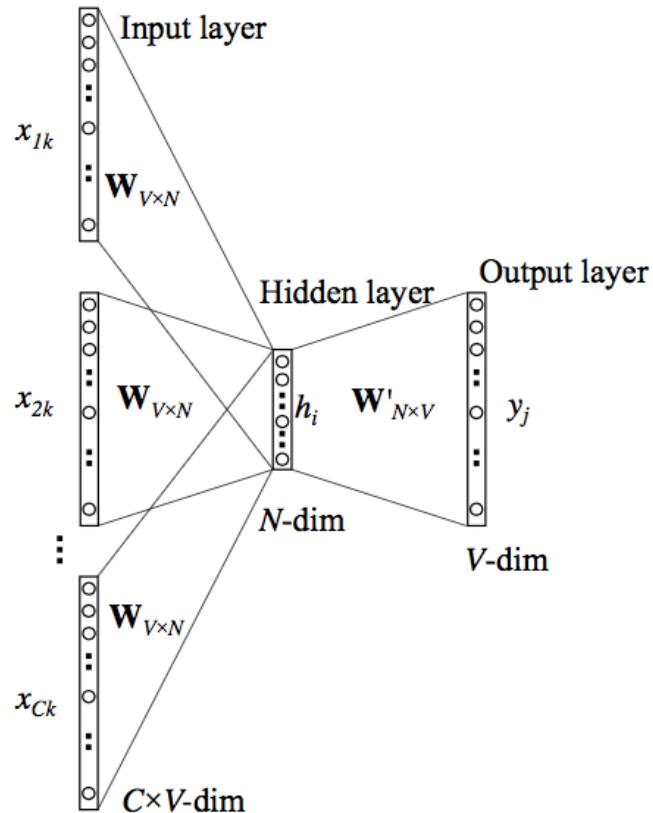


Рисунок 2.10. CBOW для декількох слів

Наведена вище модель бере слова C контексту. W_{vp} використовується для обчислення прихованих вхідних шарів, ми беремо середнє значення для всіх цих вхідних текстів у контексті C .

Отже, ми побачили, як представлення слів формуються за допомогою контекстних слів. Але є ще один спосіб зробити те саме. Ми можемо використовувати цільове слово (представлення якого ми хочемо генерувати) для прогнозування контексту, і в процесі цього ми виробляємо уявлення. Інший варіант, який називається моделлю Skip Gram, робить це.

2.2.4.2 Skip Gram

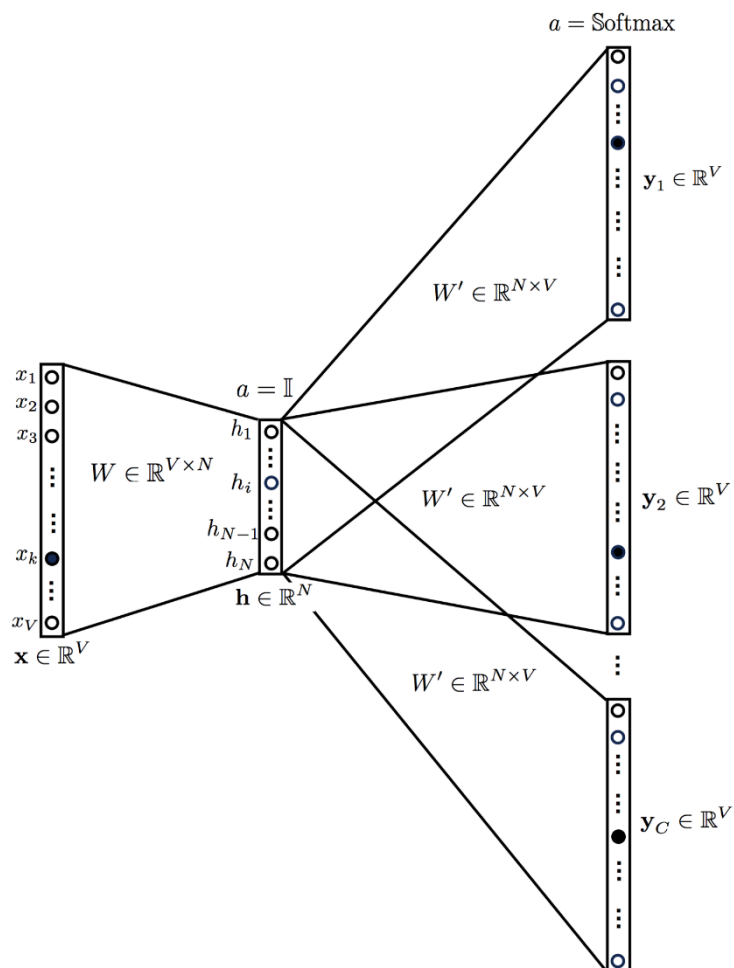


Рисунок 2.11. Skip Gram

Це виглядає так, як щойно перевернута модель з декількома контекстами CBOW. В якійсь мірі це правда.

Ми вводим цільове слово в мережу. Модель виводить розподіли ймовірностей C . Що це означає?

Для кожної позиції контексту ми отримуємо C розподіли ймовірностей V ймовірностей, по одному на кожне слово.

В обох випадках мережа використовує зворотне поширення для навчання.

У обох є свої переваги та недоліки. Skip Gram добре працює з невеликою кількістю даних і, як відомо, добре позначає рідкісні слова.

З іншого боку, CBOW швидше і має кращі подання для частіших слів.

2.3 Нейронна мережа

Проблеми передбачення послідовності існують вже давно. Вони вважаються однією з найскладніших проблем, що вирішуються в галузі інформатики. Сюди входить широке коло проблем; від прогнозування продажів до пошуку шаблонів даних фондових ринків, від розуміння сюжетів фільмів до розпізнавання способу мовлення, від перекладу мови до передбачення наступного слова на клавіатурі iPhone.

З недавніми проривами, що відбулися в науці даних, встановлено, що майже для всіх цих проблем передбачення послідовності мережі довгострокової тимчасової пам'яті, як LSTM, були визнані як найбільш ефективне рішення.

LSTM мають перевагу над звичайними нейронними мережами, що поширюються вперед і RNN, багатьма способами. Це пояснюється їх властивістю вибірково запам'ятовувати шаблони протягом тривалого часу.

2.3.1 Recurrent Neural Networks

Візьмемо приклад послідовних даних, якими можуть бути дані фондового ринку для певної акції. Проста модель машинного навчання або штучна нейронна мережа можуть навчитися прогнозувати ціни акцій, виходячи з ряду особливостей: обсягу акцій, вартості відкриття тощо. Хоча ціна акцій залежить від цих

особливостей, вона також значною мірою залежить від вартості запасів у попередні дні. Насправді для трейдера ці значення в попередні дні (або тенденція) є одним з головних визначальних факторів для прогнозів.

У звичайних нейронних мережах, що передаються вперед, всі тестові випадки вважаються незалежними. Тобто при встановленні моделі на певний день, ціни на акції за попередні дні не враховуються.

Ця залежність від часу досягається за допомогою рекуррентних нейронних мереж. Типовий RNN виглядає так:

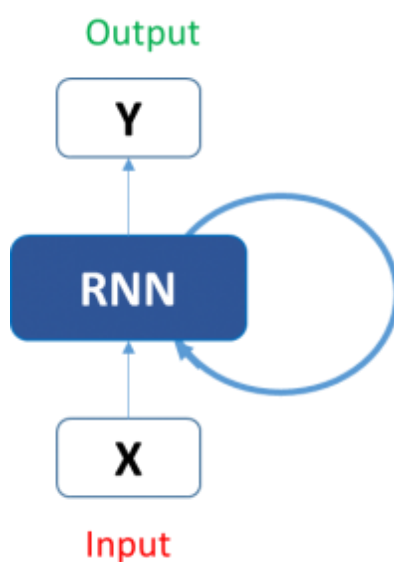


Рисунок 2.12. RNN

Це може бути залякуючим з першого погляду, але коли її розгорнути, це виглядає набагато простіше:

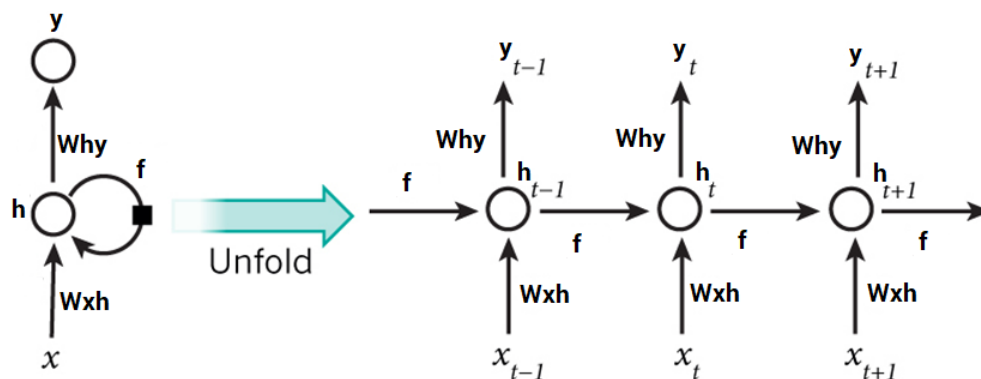


Рисунок 2.12. Розгорнута RNN

Тепер нам простіше уявити, як ці мережі розглядають тенденцію цін на акції, перш ніж прогнозувати ціни на акції на сьогодні. Тут кожне передбачення на час t залежить від усіх попередніх прогнозів та інформації, отриманої з них.

RNN можуть вирішити нашу мету обробки послідовностей значною мірою, але не повністю. RNN – це чудово, коли мова йде про короткі контексти, але для того, щоб мати можливість будувати історію і запам'ятовувати її. Але що ж робити якщо нам потрібно запам'ятати якусь інформацію на досить довгий проміжок часу. Це неможливо при простому RNN.

2.3.2 RNN на основі LSTM

Коли ми плануємо наш графік на цей день, ми визначаємо пріоритетність своїх зустрічей. Якщо нам потрібно звільнити час для чогось важливого, ми знаємо, яку зустріч можна було б скасувати.

Виявляється, RNN не робить цього. Для того, щоб додати нову інформацію, вона повністю перетворює наявну інформацію, застосовуючи функцію. Через це вся інформація в цілому модифікується, тобто не враховується "важлива" інформація та "не така важлива" інформація.

З іншого боку, LSTM вносять невеликі зміни до інформації шляхом множення та додавання. З LSTM інформація протікає через механізм, відомий як стан комірок. Таким чином, LSTM можуть вибірково запам'ятовувати або забувати речі. Інформація в конкретному стані комірки має три різні залежності.

Ми це візуалізуємо на прикладі. Візьмемо для прикладу прогнозування цін на акції для певної акції. Ціна акцій сьогодні буде залежати від:

- Тенденції, яку акції спостерігали в попередні дні, може бути, спад або зростання;
- Ціна акції за попередній день, оскільки багато торговців порівнюють ціну акцій за попередній день, перш ніж купувати її;

- Фактори, які можуть вплинути на ціну акцій на сьогодні. Це може бути нова політика компанії, яку широко критикують, або падіння прибутку компанії, або, можливо, несподівана зміна вищого керівництва компанії.

Ці залежності можна узагальнити до будь-якої проблеми як:

- Попередній стан комірки (тобто інформація, яка була присутня в пам'яті після попереднього кроку);
- Попередній прихований стан (тобто це те саме, що і вихід попередньої комірки);
- Вхід на поточному етапі часу (тобто нова інформація, яка надходить у цей момент).

2.3.3 Архітектура LSTM

Функціонування LSTM можна уявити, зрозумівши функціонування команди каналу новин, яка висвітлює історію вбивства. Тепер новина будується навколо фактів, доказів та тверджень багатьох людей. Щоразу, коли відбувається нова подія, ви робите будь-який із трьох кроків.

Скажімо, ми припускали, що вбивство було скоєне "отруєнням" потерпілого, але у звіті про аутопсію, що щойно надійшла, сказано, що причиною смерті був "удар по голові". Будучи частиною цієї команди новин, що ти робиш? Ви відразу забуваєте попередню причину смерті та всі історії, які були сплетені навколо цього факту.

Що робити, якщо в картину вводиться абсолютно новий підозрюваний. Людина, яка мала скарги на жертву і могла бути вбивцею? Ви вводите цю інформацію у свою новинну стрічку, правда?

Тепер усі ці розбиті фрагменти інформації не можна подавати на медіа. Отже, через певний проміжок часу вам потрібно узагальнити цю інформацію та вивести відповідні речі вашій аудиторії. Можливо, у формі «Х виявляється головним підозрюваним».

Тепер перейдемо до деталей архітектури мережі LSTM:

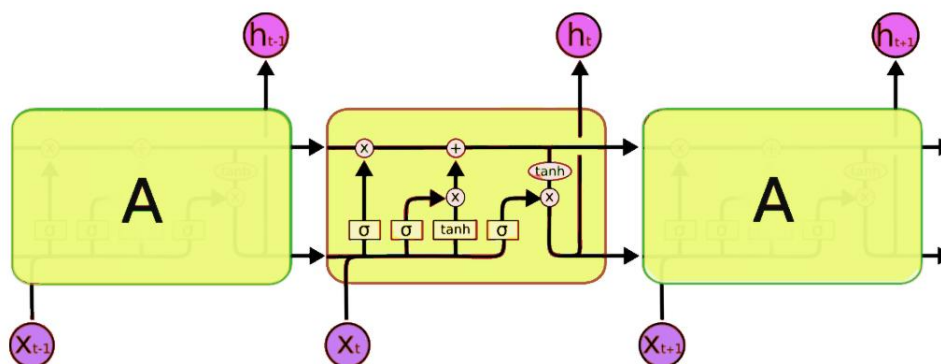


Рисунок 2.13. Архітектура мережі LSTM

Типова мережа LSTM складається з різних блоків пам'яті, які називаються осередками (прямокутники, які ми бачимо на зображенні). Є два стани, які переносяться до наступної комірки; стан клітини та прихований стан. Блоки пам'яті відповідають за запам'ятовування речей, а маніпуляції з цією пам'яттю здійснюються за допомогою трьох основних механізмів, званих воротами. Кожен з них обговорюється нижче.

2.3.3.1 Ворота забуття

Ворота забуття відповідає за виведення інформації зі стану комірки.

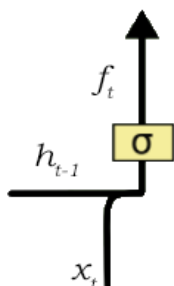


Рисунок 2.14. Ворота забуття

Інформація, яка більше не потрібна LSTM для розуміння речей, або менш важлива інформація видаляється шляхом множення фільтра. Це потрібно для оптимізації продуктивності мережі LSTM.

Ця частина приймає два входи: h_{t-1} і x_t .

h_{t-1} – прихований стан з попередньої комірки або вихід попередньої комірки, а x_t – вхід на цьому конкретному етапі часу. Дані введення множать на вагові матриці і додається зміщення. Після цього сигмоїдна функція застосовується до цього значення. Сигмоїдна функція виводить вектор, значення якого варіюються від 0 до 1, що відповідає кожному номеру в стані комірки. В основному, сигмоїдна функція відповідає за рішення, які значення зберігати, а які відкидати. Якщо для певного значення в стані комірки виводиться "0", це означає, що ворота забуття хочуть, щоб стан комірки повністю забув цю інформацію. Аналогічно, "1" означає, що ворота забуття хочуть запам'ятати всю цю інформацію. Цей векторний вихід із сигмоїдної функції множиться на стан комірки.

2.3.3.2 Вхідні ворота

Вхідні ворота відповідають за додавання інформації до стану комірки. Це доповнення інформації – це в основному триетапний процес, як видно з рисунку нище.

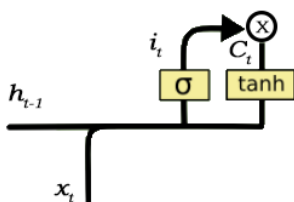


Рисунок 2.15. Вхідні ворота

1. Регулювання того, які значення потрібно додати до стану клітини, за допомогою сигмоїдної функції. Це в основному дуже схоже на ворота забуття і діє як фільтр для всієї інформації з h_{t-1} і x_t .
2. Створення вектора, що містить усі можливі значення, які можна додати у стан комір. Це робиться за допомогою функції \tanh , яка виводить значення від -1 до +1.
3. Перемноження значення регуляторного фільтра (сигмоїдний затвор) на створений вектор (функція \tanh), а потім додавання цієї корисної інформації до стану комір за допомогою операції додавання.

Після того, як цей триетапний процес буде виконано, ми гарантуємо, що до стану клітини додається лише така інформація, яка є важливою і не є зайвою.

2.3.3.3 Вихідні ворота

Функціонування вихідних воріт може бути розбито на три етапи:

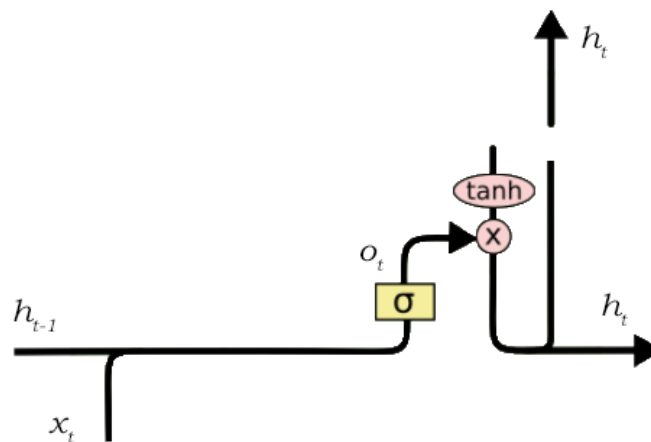


Рисунок 2.16. Вихідні ворота

1. Створення вектора після застосування функції \tanh до стану клітини, тим самим масштабування значень до діапазону від -1 до +1.

2. Створення фільтра з використанням значень h_{t-1} і x_t таким чином, що він може регулювати значення, які потрібно виводити з вектора, створеного вище. Цей фільтр знову використовує сигмоподібну функцію.
3. Множення значення цього регуляторного фільтра на вектор, створений на кроці 1, і відправлення його як вихід, а також у прихований стан наступної комірки.

2.4 Процес навчання

Розглянемо послідовність обробки даних при навчанні. Коли дані надходять до nlr, спочатку відбувається виділення тексту для отримання Doc об'єкта. Потім обробляється в декількох різних кроків – що називається конвеєром обробки, або ж трубопроводом. Трубопровід, який використовується, складається з тегера, аналізатора та розпізнавача сутності, тощо. Кожен компонент трубопроводу повертає оброблений Doc, який потім передається наступному компоненту.

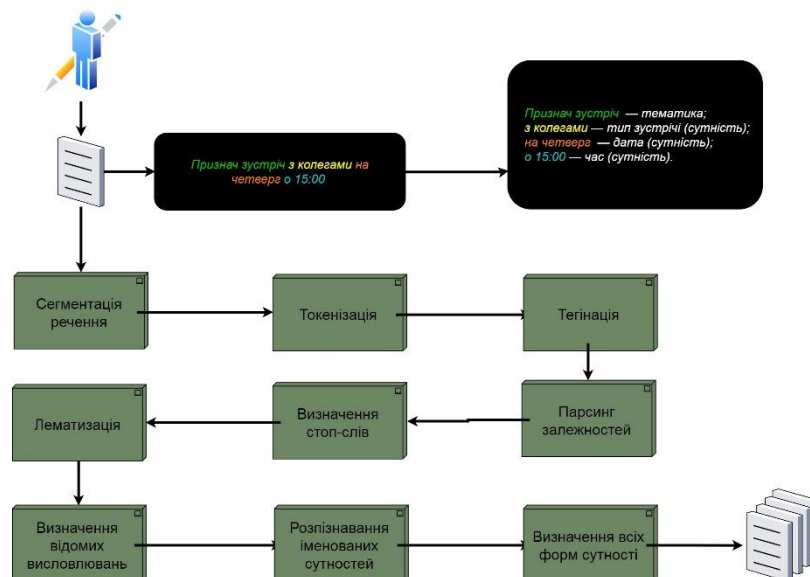


Рисунок 2.17. Конвеєр обробки тексту

1. На етапі сегментації йде розбиття на частини окремі речення, фрази, або ж слова.
2. Токенізація – розбиті на частини речення проходять етап, на якому вони відмічаються певні сутності, або ж токени.
3. Тегінація – процес схожий до токенізація, але тут йде позначення слів як частин мови, у той час як токенами може бути що завгодно.

4. Лематизація — перетворення слова в базову форму, або ж лемму. У більшості випадків, це дозволяє зберегти загальний сенс речення, достатній для розуміння машиною, але при цьому значно прискорити обробку.
5. Визначення стоп-слів — знаходження символів, які є синтаксичним кінцем речення, абзацу, тексту, або ж слів, які є синтаксичним кінцем фрази, тощо.
6. Впровадження зв'язків між словами та визначення прямих і непрямих залежностей між ними.
7. Визначення фраз, або словосполучень, які є широко відомими, або шукаються для специфічних цілей програми.
8. Розпізнавання іменованих сутностей.
9. Coreference resolution — це завдання знайти в тексті всі вирази, що відносяться до тієї ж сутності. Наприклад, між сутністю «інтелектуальний агент» та його згадуваннями в тексті такими як, «він», «агент», тощо.

Після повної обробки всі слова приводяться до векторного простору та подаються на вхід нейронної мережі.

Висновки до розділу

В даному розділі детально розглянуто функціональну схему інтелектуального агента, описаний процес взаємодії з користувачем, з наведенням конкретних прикладів. Описаний набір даних для навчання, детальний процес його формування, обґрунтування вибору. Проаналізовані нейронні мережі, та обрано оптимальну з існуючих. Описаний процес навчання.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Архітектура програмного забезпечення

У попередньому розділі ми вже детально розглянули функціональну схему системи. Зараз же розберемо архітектуру програмного забезпечення на якому вона побудована. Для кращого розуміння візьмемо конкретний випадок:

1. Надання інформації. Користувач вводить повідомлення використовуючи для цього текстовий, або голосовий спосіб.
2. Конвертація даних. У залежності від того через від способу введення необхідними є відповідні апаратні та програмні засоби. Наприклад, у більшості ситуацій користувацькі пристрої підтримують текстовий формат повідомлень, але зараз вже більш популярними стають інтелектуальні консультанти, на зразок тих, які використовуються в аеропортах, і деякі з них підтримують тільки голосове ведення. Іншим прикладом, де підтримуються тільки аудіо взаємодія, є роботи-співрозмовники.
3. Після початкового апаратного рівня інформація передаються до програми, за допомогою якої здійснюється взаємодія. Це можуть бути додатки типу WEB, мобільних, або на будь-якій платформі ОС. У нашому випадку, набагато вигіднішим є інтеграція в вже існуючі месенджери, так як є це гарантує прийняття нововведень користувачами, позбавляє необхідності встановлення, або налаштування та є дуже економічно вихідним.
4. Далі йде передача даних на фізичний сервер, на якому і знаходить головна API, або який є зв'язковою ланкою для обміну даними. Перший варіант є пріоритетнішим, так як у цьому випадку прискорюється швидкість взаємодії з користувачем, що є одним з головних параметрів при розробці подібних систем.

5. У самій API власне й знаходиться майже вся логіка програми, яку ми вже трішки розглядали вище, і ще розглянемо детальніше в наступних підрозділах. З точки зору апаратної реалізації нас цікавить взаємодія з іншими пристроями, і вона визначається саме тут, при обробці типу повідомлення.
6. Ці модулі можуть бути найрізноманітнішими і залежать часто від конкретного спектру задач, які потрібно виконати. Дуже часто використовується база даних, гугл календарь, гугл карти та багато іншого. Це може бути й виконання якихось механічних дій, наприклад, у випадку з роботом, що буде переміщуватися повністю, або рухати якісь частини.

3.2 Взаємодія з користувачем

Процес користування є простим та інтуїтивно зрозумілим для кожного. Він може трішки відрізнятись у залежності від платформи в яку він інтегрований, але ці зміни є зовсім не значними.

У загальному випадку інтерфейс виглядає так:

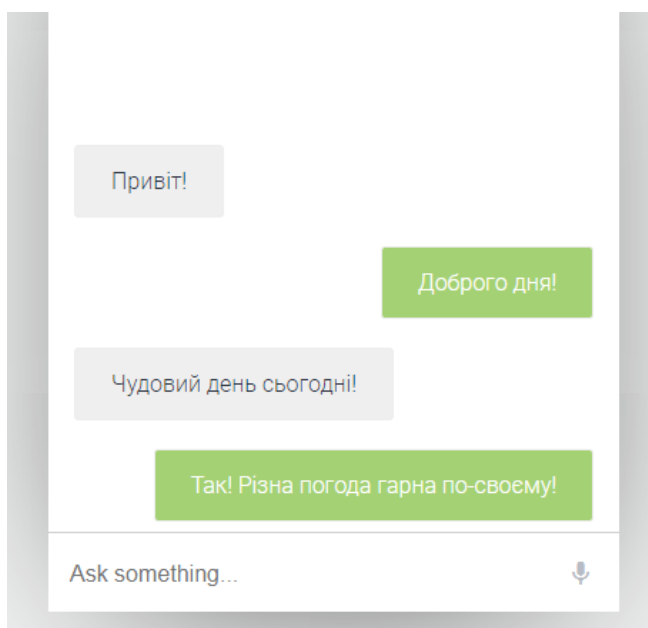


Рисунок 3.1. Загальний приклад роботи

Інтерфейс є мінімалістичним та комфортним для користування. Вікно взаємодії розділено на дві частини:

1. Історія розмови.
2. Панель введення.

Остання ж у своє чергу ділиться на:

1. Поле введення текстової інформації.
2. Кнопки для активації можливості голосового надання даних.

Все це робить процес взаємодію дуже зручним для користувача.

Одна з найголовніших переваг інтелектуальних агентів, це гнучкість у роботі. Так, наприклад, якщо ви зовсім не знайомі з помічником, ви можете напряду про це у нього запитати і він вам допоможе!

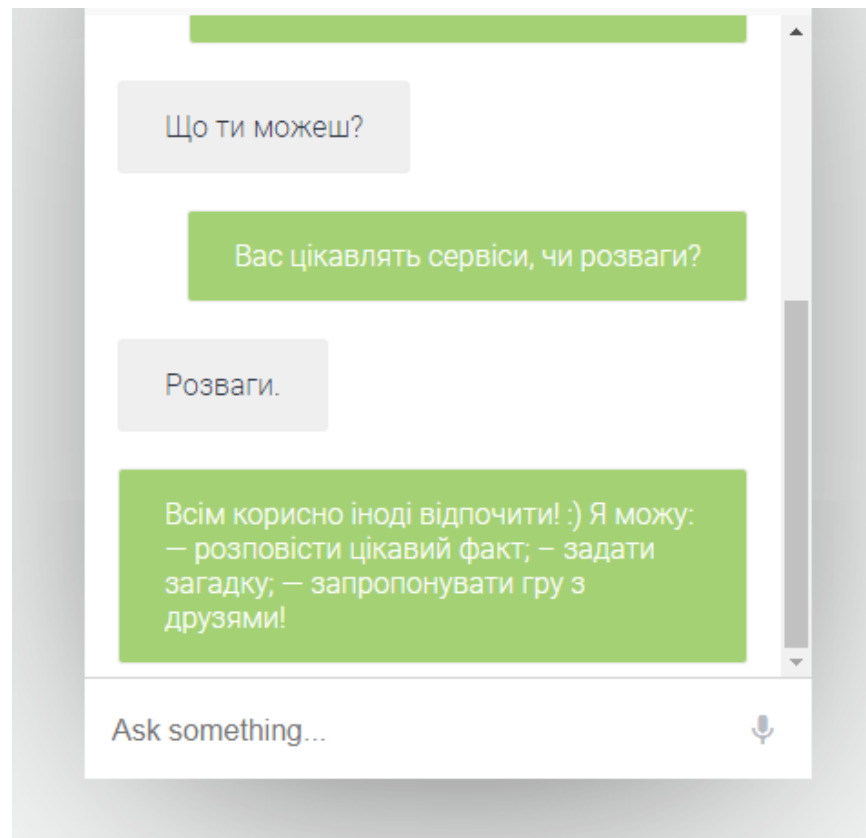


Рисунок 3.2. Приклад надання довідкової інформації

3.3 Алгоритм роботи

Розглянемо детально алгоритм роботи інтелектуального агента з наведенням конкретних прикладів.

1. Користувач вводить повідомлення текстом, або надає інформацію у аудіо форматі.
2. Детальніше процес отримання, конвертації даних за допомогою програмних та апаратних засобів був описаний раніше. Загалом ця частина називається Connector Modules.
3. Потім дані потрапляють у модуль обробки. Тут інформація проходить ряд перетворень під час якою виділяються потрібні дані та транслюються у формат зручний для нейронної мережі.
 - a. У цей блок входять, модулі які використовуються майже завжди, як стандарт, такі як тегер, парсер, розпізнавач сутностей.
 - b. Також є обробники, використання яких характерно тільки для певних груп інтелектуальних агентів, наприклад лематизація
 - c. Є й модулі специфічні для даної моделі. Це й сегментація на початку, тегінація частин мови, знаходження стоп слів, розпізнавання відомих висловлювань, що є надзвичайно важливим для розуміння вільної мови.
4. Після попередньої обробки повідомлення транслюється у формат bag of words.
5. Векторне представлення повідомлення порівнюється з векторним представленням тем, доступних для даного інтелектуального агента.
6. Завдяки тому, що рекурентна нейронна мережа оснований на LSTM-блоках вдається зберігати і використовувати контекст розмови.
7. Після обробки нейронною мережею користувач отримує відповідь і виконуються якісь програмні дії, якщо вони потрібні. Наприклад, звернення до бази даних або виклик іншого API.

Для кращого розуміння та наочності, скористаємось доступною функціональністю розробленої системи.

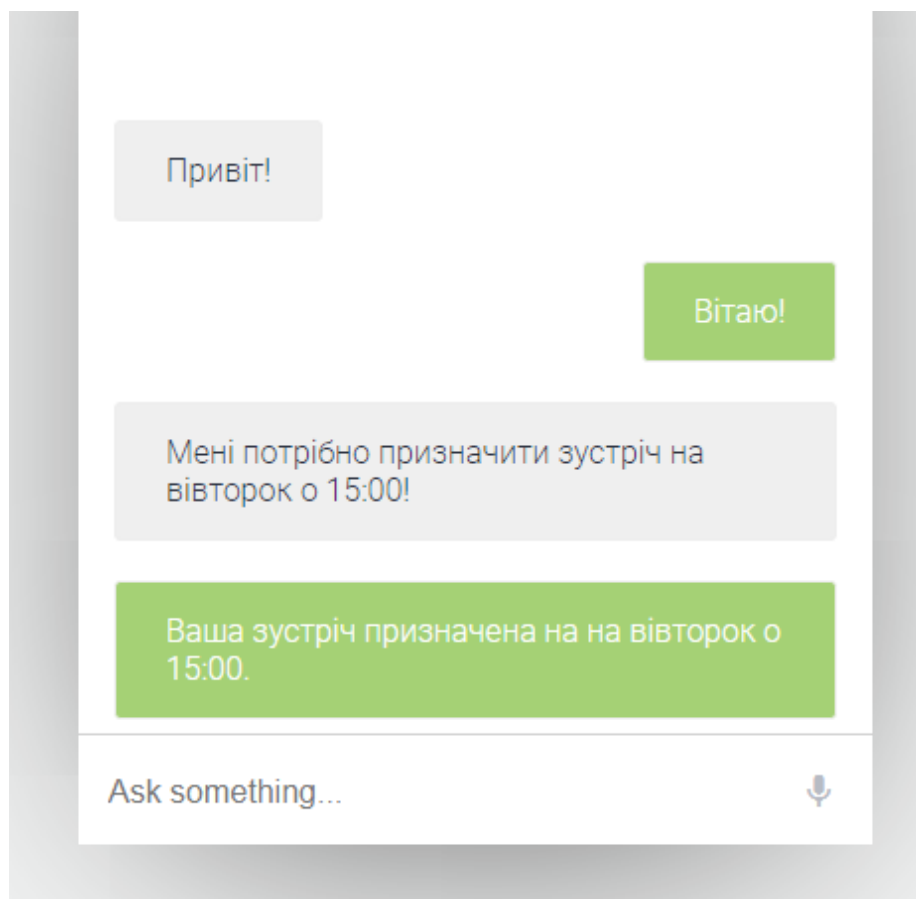


Рисунок 3.3. Приклад сервісу

У даному випадку ми надали всю потрібну інформацію, потрібну для здійснення операції, тому вона відразу була успішно виконана. При цьому була здійснення взаємодія з іншим API, завдяки якому дані були занесені до гугл календаря і у випадку спроби призначення зустрічі на цей час ще раз, буде отримано відмову і запропоновано обрати інший час.

Також, даний функціонал досить просто розширити. Наприклад, у даної теми може бути обов'язкова сутність «тип зустрічі», яка буде зберігатися у базі даних. У цьому випадку, вона є не обов'язковою.

Тепер, спробуємо надати не всю інформацію.

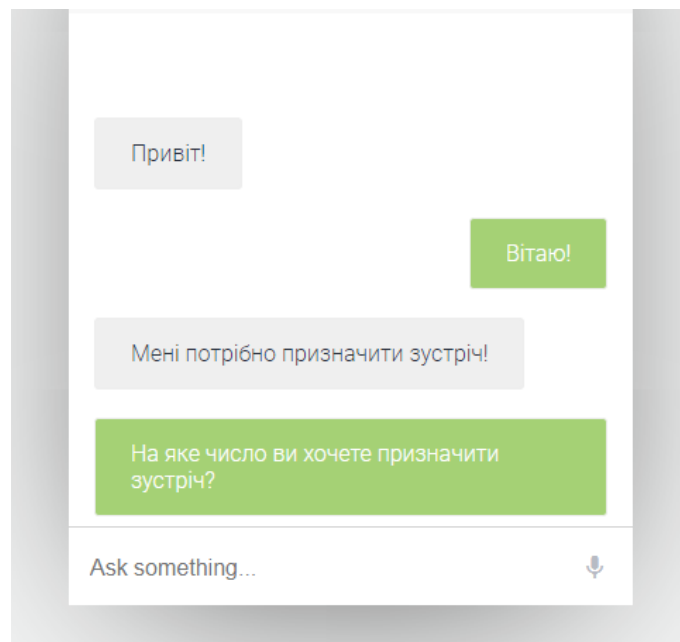


Рисунок 3.4. Приклад неповної інформації

У результаті обробки тема була успішно розпізнана, але для виконання операцій, зв'язаних з нею, не було знайдено потрібних даних, тому інтелектуальних агент задає уточнюючі запитання.

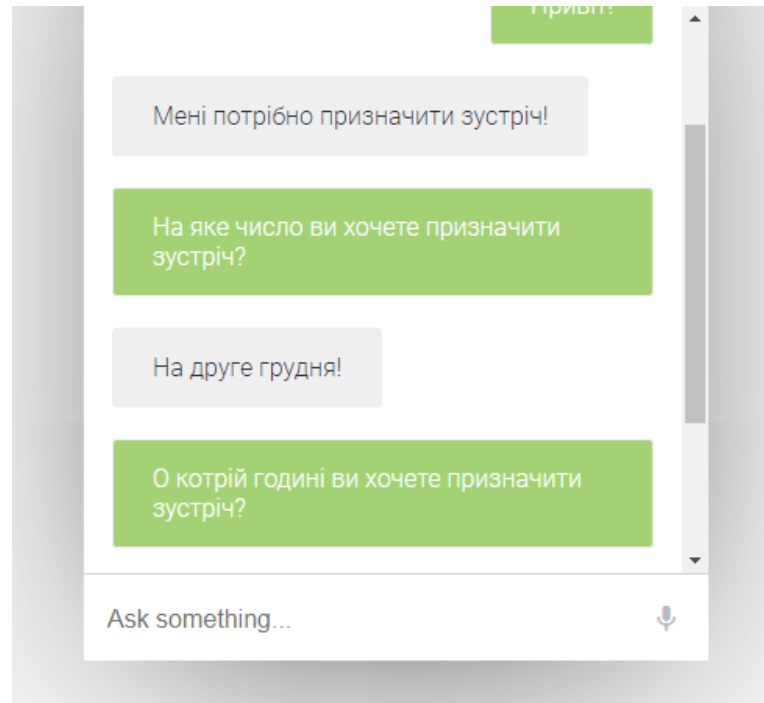


Рисунок 3.5. Приклад уточнення інформації

Ми зробили своє прохання конкретнішим, але інформація все ще не повна.

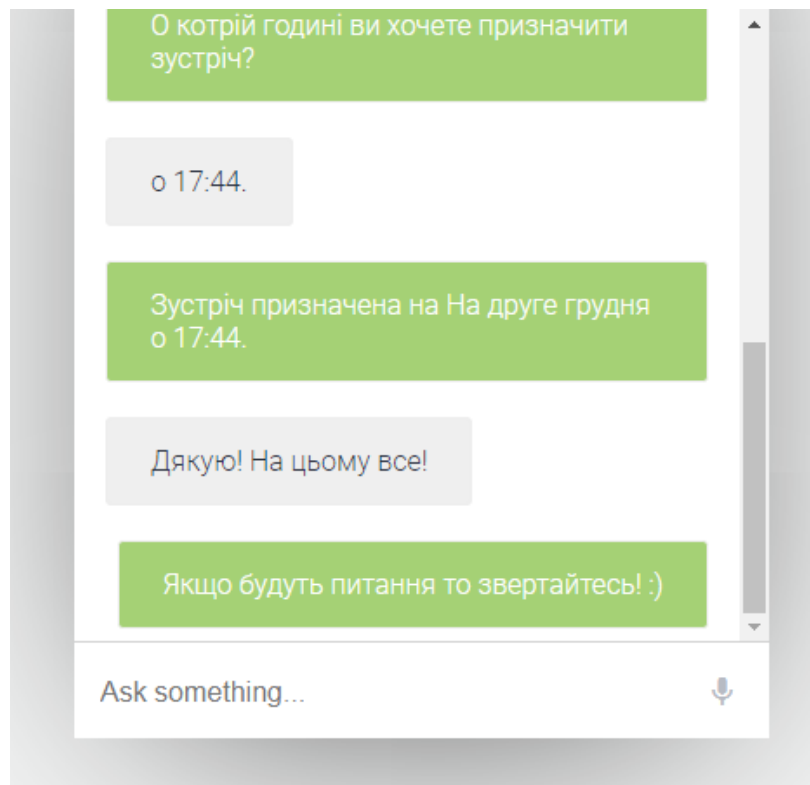


Рисунок 3.6. Приклад закінчення розмови

Як бачимо, у даній темі було 2 сутності і вони обидні були відновленні інтелектуальним агентом, що дало можливість вдало закінчити операцію.

3.4 Вибір засобів для розробки

Як основний програмний засіб для розробки був обраний TensorFlow. Є ряд причин чим це зумовлено:

1. Простота побудови моделі. TensorFlow пропонує кілька рівнів абстракції, щоб ми могли вибрати правильний для наших потреб. Дає можливість створювати та навчати моделі за допомогою API Keras високого рівня, що робить роботу з TensorFlow та машинним навчанням досить простою.
2. Надійна продукція ML в будь-якому місці. TensorFlow завжди забезпечував прямий шлях до виробництва. Будь то на серверах, периферійних пристроях або в Інтернеті, TensorFlow дозволяє легко навчати та розгортати модель, незалежно від мови та платформи, якими ви

користуєтесь. Для отримання повного конвеєру ML існує TensorFlow Extended (TFX). Для виконання на мобільних та периферійних пристроях існує TensorFlow Lite. Для розгортання моделей в середовищах JavaScript, використовують TensorFlow.js.

3. Потужний засіб для досліджень. Створення та тренування найсучасніших моделей, не жертвуючи швидкістю та продуктивністю. TensorFlow надає гнучкість та контроль за допомогою таких функцій, як Keras Functional API та API субкласифікації моделей для створення складних топологій. TensorFlow також підтримує екосистему потужних додаткових бібліотек та моделей, з якими можна експериментувати, включаючи Ragged Tensors, TensorFlow Probability, Tensor2Tensor та BERT.

Іншим програмним засобом є Keras. API нейронних мереж високого рівня, написаний на Python і здатний працювати над TensorFlow, CNTK або Theano. Він був розроблений з акцентом на швидке експериментування. Вміння переходити від ідеї до результату з найменшою можливою затримкою є ключовим для хорошого дослідження.

Це бібліотека глибокого навчання, яка:

- Дозволяє легко та швидко прототипувати (завдяки зручності користування, модульності та розширюваності);
- Підтримує як зготкові мережі, так і періодичні мережі, а також їх комбінації;
- Працює безперебійно на процесорі та GPU.

Ще одним програмним засобом, який застосовувався, є додаток розроблений мною особисто на мові Java. Він використовувався для часткового формування набору даних, шляхом обробки інформації на різних сервісах з підтримки користувачів.

3.5 Вимоги до технічного та програмного забезпечення

- процесор – 1.6 ГГц, 1 ядро ЦП або краще;
- оперативна пам'ять не менш ніж 1024 Мб;

- не менше ніж 35 ГБ ПЗУ;
- доступ до мережі Інтернет;
- операційна система:
 - Linux,
 - Mas OX,
 - Windows, and others.

3.6 Результати роботи

Конкретні результати роботи з прикладами були розглянуті вище, у цьому ж розділі. У цьому підрозділі описані результати експериментального дослідження. Моделлю на, якій функціонує нейронна мережа є модель bag of words. Вона є основою для функціонування системи. Задачею є визначення параметрів, при яких функціонування мережі буде найоптимальнішим.

Першим досліджуваним параметром був — кількість фраз, які входять у training phrases для кожної теми. Дослідження проводиться для моделі яка навчена на 5000 фраз.

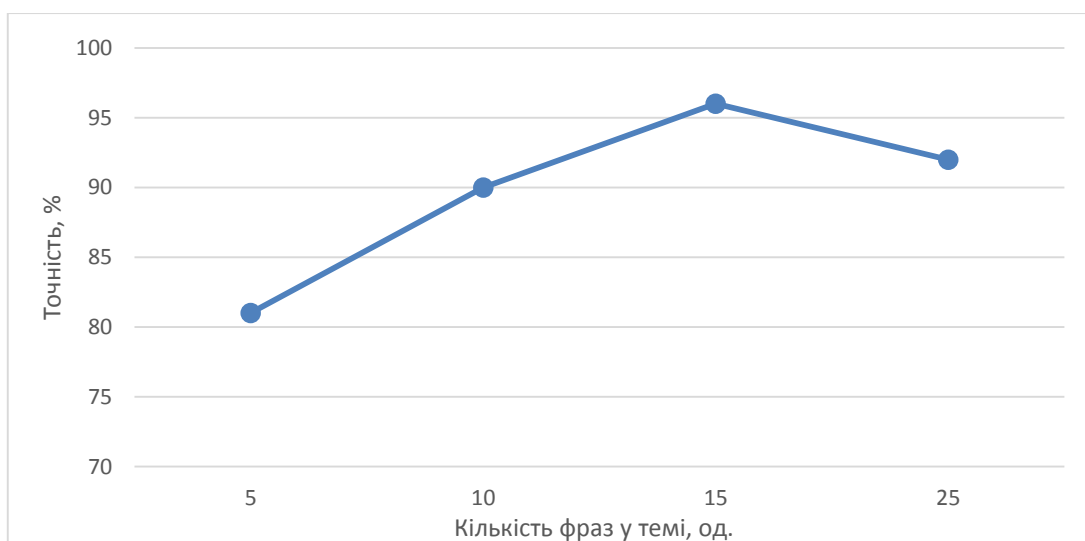


Рисунок 3.7. Залежність точності від кількості навчальних фраз у темі.

Наступним досліджуваним параметром — кількість слів, які входять до однієї фрази. Дослідження проводиться для моделі яка навчена на 5000 фраз.

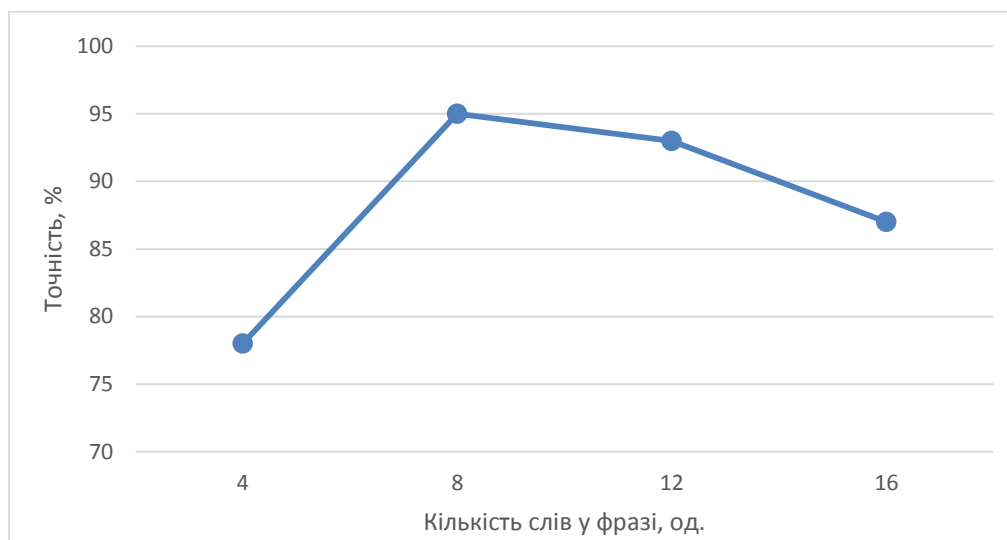


Рисунок 3.8. Залежність точності від кількості слів у фразі.

Наступним досліджуваний параметр — зміна кількості слів, які входять до однієї фрази при зміні об'єму даних для навчання. Точність зберігається у районі 95%.

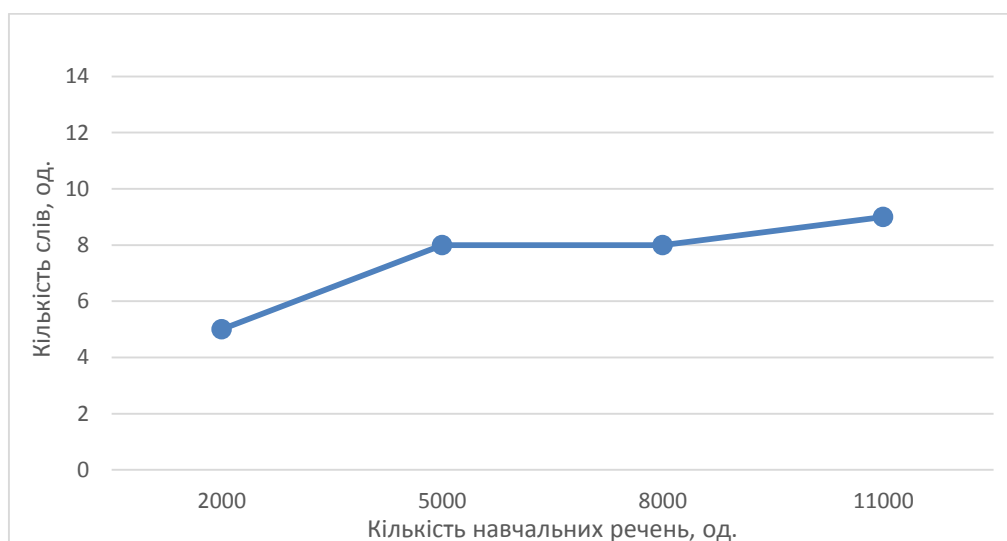


Рисунок 3.9. Залежність кількості слів, які є оптимальними для однієї фрази при зміні об'єму даних для навчання.

Наступним досліджуваний параметр — зміна кількості фраз при зміні об'єму даних для навчання. Точність зберігається у районі 95%.

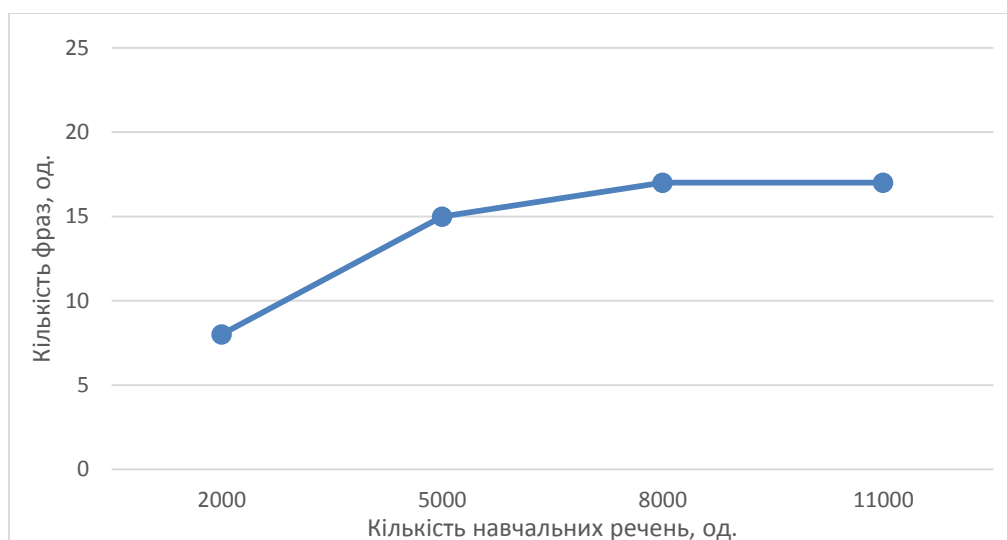


Рисунок 3.10. Залежність зміни кількості фраз при зміні об'єму даних для навчання.

Дослідження різних параметрів показало, що при малому наборі даних для навчання точність погіршується через недостатнє розуміння тематики питання, а при надмірному кількості фраз відбувається так званий *overfitting*, або перенавчання. Було визначено, що для моделі *bag of words* оптимальними параметрами є: кількість фраз, які входять у *training phrases* — 15, кількість слів, які входять до однієї фрази — 8. І також, те що при збільшенні об'єму даних для навчання корегування параметрів є не значним.

Висновки до розділу

У даному розділі було детально розглянуто архітектуру розробленої системи, описані етапи роботи алгоритмічного забезпечення, визначені вимоги до технічного забезпечення, описані програмні засоби, що використовувалися. Зображено процес взаємодії з користувачем та інтерфейс. Підведені підсумки по результатах проведеної роботи.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Узагальнено етапи розроблення стартап-проекту можна подати таким чином:

1) Маркетинговий аналіз стартап-проекту

В межах цього етапу:

- розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються ринкові можливості щодо його реалізації;
- на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

2) Організація стартап-проекту

В межах цього етапу:

- складається календарний план-графік реалізації стартап-проекту;
- розраховується потреба в основних засобах та нематеріальних активах;
- визначається плановий обсяг виробництва потенційного товару, на основі чого формулюється потреба у матеріальних ресурсах та персоналі;
- розраховуються загальні початкові витрати на запуск проекту та планові загальногосподарські витрати, необхідні для реалізації проекту.

3) Фінансово-економічний аналіз та оцінка ризиків проекту

В межах цього етапу:

- визначається обсяг інвестиційних витрат;

- розраховуються основні фінансово-економічні показники проекту (обсяг виробництва продукції, собівартість виробництва, ціна реалізації, податкове навантаження та чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності проекту);
- визначається рівень ризикованості проекту, визначаються основні ризики проекту та шляхи їх запобігання (реагування на ризики).

4) Заходи з комерціалізації проекту. Цей етап спрямовано на пошук інвесторів та просування інвестиційної пропозиції (оферти). Він передбачає:

- визначення цільової групи інвесторів та опису їх ділових інтересів;
- складання інвест-пропозиції (оферти): стислої характеристики проекту для попереднього ознайомлення інвестора із проектом;
- планування заходів з просування оферти: визначення комунікаційних каналів та площадок та планування системи заходів з просування в межах обраних каналів;
- планування ресурсів для реалізації заходів з просування оферти.

Означені етапи, реалізовані послідовно та вчасно – створюють передумови для успішного ринкового старту.

4.1 Опис ідеї проекту

Зміст ідеї проекту, основні напрямки застосування та їх вигоди для користувача наведено у табл. 4.1.

Проект відрізняється від аналогів тим що:

1) Для реалізації використовуються українська мова. На основі української мови не існує жодного відомого чат-бота. Його створення дасть можливість подальшого розвитку у цьому напрямку обробки вільної мови. Також, ця

особливість робить можливим застосування подібних чат-ботів для використання в сфері бізнесу та послуг.

2) Працює на основі нейронної мережі, що дає можливість обробки вільної мови, а не конкретних вказівок, як це реалізовано у багатьох аналогів. Ще більш зручним і актуальним це стає у зв'язку з тим, що подання інформації можна здійснювати у голосовому форматі, який потім конвертується у текст і оброблюється.

3) Чат-бот легко інтегрується у безліч існуючих месенджерів, якими, за результатами офіційних опитувань, користується більше 80% користувачів мобільних телефонів. Це позбавляє необхідності створення окремої платформи для інтеграції програмного продукту, і як результати, великої кількості витрат. За рахунок того, що месенджери, у які інтегрований чат-бот є дуже популярними, гарантується велика ймовірність прийняття нововведення. Також, це позбавляє користувача необхідності встановлювати щось, або налаштовувати.

4) Розроблений інтерфейс програми є інтуїтивно зрозумілим та дружнім для користувача.

5) Особлива реалізація функціональності, підготовки та обробки даних, про яку детально розповідалось раніше, робить чат-бота ефективнішим у порівнянні з багатьма аналогами.

6) Загалом всі вище перелічені фактори роблять використання даного програмного продукту надзвичайно зручним та ефективним.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка інтелектуального агента, здатного розуміти та коректно обробляти запити користувача, використовуючи вільну українську мову.	Напрямки для застосування розробленої моделі є без перебільшення безмежними. Так як вона залежить від даних для навчання і внутрішніх налаштувань, які можуть здійснюватися у порівняно короткий час. Для прикладу, застосування може починатися сферою навчання та закінчуватися сферою бізнесу і надання послуг.	Користувача отримує ряд значних вигод: <ul style="list-style-type: none"> • Зручний спосіб отримання послуг. • Інтеграція у всіх популярних месенджерах. • Відсутність необхідності встановлення, або налаштування. • Можливість звернення вільною мовою. • Можливість голосових повідомлень.

При розгляді конкурентів спробуємо охопити весь наявний ринок шляхом правильного виробу кандидатів, які будуть у повному обсязі характеризувати спектр наявних можливостей. Трьох кандидатів буде достатньо:

- Roof AI
- Realbot Chatbot
- Google assistant

Таблиця 4.2. Опис ідеї стартап-проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Roof AI	Realbot	Google assistant			
1	Точність відповідей	Висока	Висока	Висока	Висока		+	
2	Кросплатформеність	Так	Ні	Ні	Так			+
3	Обробка голосових повідомлень	Так	Ні	Ні	Так		+	
4	Зручність використання	Висока	Середня	Середня	Висока			+
5	Необхідність встановлення	Ні	Так	Так	Ні			+
6	Швидкість обробки	Висока	Середня	Середня	Висока			+
7	Обмеженість сферою застосування	Так	Так	Так	Ні		+	
8	Використання нейромережі	Так	Ні	Ні	Так			+
9	Можливість перенавчання	Так	Ні	Ні	Ні			+
10	Можливість спілкуватися українською	Так	Ні	Ні	Ні			+
11	Комерційне застосування	Так	Так	Так	Ні		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї проекту

Проведемо аудит технологій, за допомогою яких можна реалізувати ідею проекту та визначимо технологічну здійсненність ідеї проекту.

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення WEB-додатку	HTML, CSS, JavaScript, React.js, Python, MySQL DB, Keras, Tensorflow	+	+
2	Мобільний додаток	HTML, CSS, JavaScript, Java, Python, MySQL DB, Keras, Tensorflow, C#, C++	+	+
3	Програмне забезпечення для ОС	HTML, CSS, JavaScript, Java, Python, MySQL DB, Keras, Tensorflow, C#, C++	+	+
4	Частина програми, що широко використовується	HTML, CSS, JavaScript, Java, Python, MySQL DB, Keras, Tensorflow, можливість проведення інтеграції	+	+
<i>Обрана технологія реалізації ідеї проекту: 4</i>				

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №4. Причини, які зумовили виріб 4 варіанту:

- Економічна доцільність.
- Велика ймовірність прийняття суспільством.

- Зручність користування.
- Можливість широкого використання для бізнесу.
- Наявність механізмів для інтеграції.

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

Визначимо ринкові можливості, які можна використати під час ринкового впровадження проєкту, та ринкових загроз, які можуть перешкодити реалізації проєкту.

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	10 (Найбільш відомих, сильно залежить від сфери застосування)
2	Загальний обсяг продаж, грн./ум.од	200000 ум.од для конкретної реалізації
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі або по ринку, %	70%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба у спілкуванні	Самотні люди, люди з обмеженими можливостями	Люди з цієї групи потребують чат-бота, як співрозмовника	Головними вимогами до товару для всіх сфер застосування та для всіх користувачів є доступність, зручність використання, точність отримуваних відповідей та швидкість їх надання.
2	Потреба у розвагах чи відпочинку	Всі категорії людей, окрім дітей до 7 років та літніх людей після 60 років	Клієнти орієнтовані на інтелектуального агента, як на спосіб розваг, конкурсів, цікавих новин, фактів, тощо	
3	Потреба у наданні послуг	Всі категорії	Користувачі очікують від чат-бота отримання певних послуг, як правило, зв'язаних з бізнесом	

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів

Закінчення таблиці 4.6

3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії
---	---------------	--	---

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи кешування даних.
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим заміником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів заміників; Різні типи ліцензій.
2	Рівень конкурентної боротьби: світовий	Всі продукти заміники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.

Закінчення таблиці 4.8

5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Додатки для бізнесу	Розважальні чат-боти	Відсутні	Майже будь-яка галузь	Існуючі рішення чат-ботів
Висновки	Всі відомі бізнес додатки організовані на основі команд, для більшої надійності, тому використання вільної мови, може бути кроком вперед.	Не існує жодного комерційно застосованого чат-бота для розваг. Що є стимулом для подібної розробки.	-	Так як створений інтелектуальний агент легко інтегрується в популярні месенджери він є зручним і може надавати послуги для будь-якої сфери підприємницької діяльності.	Існуючі рішення поки що не знайшли широкого бізнес застосування та потребують вдосконалення.

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Кросплатформеність	Можливість недорогої розробки, яка дасть можливість розміщення програмного продукту на всіх існуючих платформах.
2	Ціна додатку	Додаток є безкоштовним для користування та оплачується лише підприємцями, чи організаціями, для яких це є економічно вигідно.
3	Зручність користування	Наявність зручного на інтуїтивно зрозумілого інтерфейсу для користувача.
4	Швидкий початок роботи	Продукт не потребує жодних дій щодо встановлення чи налаштування.
5	Точність роботи	Як показали результати дослідження програмний продукти має більш ніж задовільну точність роботи.
6	Кількість реклами	Реклама повністю відсутня, що є привабливим для користувача.
7	Широта асортименту функцій	Функціонал обмежуються сферою застосування та вимогами замовника.

Закінчення таблиці 4.10

8	Висока інтегрованість	Можливість інтеграції різні популярні додатки. Переважно месенджери.
9	Новизна	Перший інтелектуальний агент, який працює використовуючи українську мову.
10	Використання нейронної мережі	Дає значну перевагу над аналогами, які використовують командний інтерфейс.
11	Обробка голосових повідомлень	Дозволяє звернення голосом, що робить зручним використання даного додатку у багатьох ситуаціях
12	Розпізнавання вільної мови	Є ще однією значною перевагою над конкурентами підвищуючи комфорт при роботі з програмним продуктом.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін системи

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Кросплатформеність	15			+				
2	Ціна додатку	18			+				
3	Зручність користування	15			+				
4	Швидкий початок роботи	15		+					
5	Точність роботи	14				+			
6	Кількість реклами	20			+				
7	Широта асортименту функцій	10						+	
8	Висока інтегрованість	15			+				
9	Новизна	10		+					
10	Використання нейронної мережі	10			+				
11	Обробка голосових повідомлень	10			+				
12	Розпізнавання вільної мови	14		+					

За визначеними факторами конкурентоспроможності було проведено аналіз сильних та слабких сторін стартап проекту

Таблиця 4.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Простота у використанні – Наявність відкритого вихідного коду – Збереження приватності інформації користувача. – Кросплатформеність – Ціна додатку – Зручність користування – Швидкий початок роботи – Точність роботи – Кількість реклами – Висока інтегрованість – Новизна – Використання нейронної мережі – Обробка голосових повідомлень – Розпізнавання вільної мови 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Широта асортименту функцій – Необхідність навчання під конкретну галузь застосування
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Застосування для будь-якої галузі – Підтримка користувачів – Розширення функціоналу – Вдосконалення алгоритму обробки даних 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – Створення схожого програмного продукту однією з світових компаній

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	1-2 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні.	1-3 місяці

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Будь-які види підприємницької діяльності	Висока	Середній	Дуже низька	Висока
2	Розважальні платформи	Висока	Середній	Середня	Висока
3	Звичайні користувачі	Висока	Середній	Середня	Висока
Які цільові групи обрано: 1, з можливістю використання групою 3					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є будь-які сфери підприємницької діяльності та надання послуг, так як даний продукт є надзвичайно гнучким у цьому плані. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію цільового маркетингу, оскільки для підприємств створюється вузько-спрямований продукт з можливістю розширення функціональності за домовленістю.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, контакт напряду з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія заняття конкурентної ніші

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Зручність	Диференціації	Відсутність необхідності встановлення, або будь-якого налаштування.	Розуміння вільної мови. Можливість інтеграції в більшість популярних існуючих платформ.
	Цінова доступність	Конкурентної ніші	Базова стратегія впровадження передбачає безкоштовне користування.	Доступність для користувачів та доцільність використання для бізнесу.
3	Кількість реклами	Конкурентної ніші	У даному програмного продукті не планується використання реклами.	Відсутність маркетингових пропозицій, роботи додаток привабливим для клієнтів.
4	Новизна	Диференціації	Створення першого інтелектуального агенту на основі української мови.	Використання української мови.

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Простий початок роботи	Майже у всіх випадках не потрібно нічого встановлювати.	Приступити до роботи з додатком можна буквально за пару хвилин.
2	Зручність використання	Використання вільної мови, як способу надання інформації	Завдяки використанню нейронної мережі та додаткового функціоналу взаємодія можлива у вільній формі та за допомогою голосового введення.
3	Цінова вигода	Користуватися додатком можна абсолютно безкоштовно.	Фінансування надходить від приватних підприємців, або корпорацій, для яких це є економічно вигідним.
4	Доступність	Впровадження майже до будь-якого популярного месенджера	Програмним забезпеченням для листування користуються всі, що робить інтелектуального агента надзвичайно комфортним у користуванні.

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система інтелектуального агента для розуміння вільної мови та надання певного роду послуг		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Зручність	Нм	Е
	Швидкість роботи	Нм	Тх
	Оптимізація	Нм	Тх
	Ціна	Нм	Е
	Документація	Нм	Тл

Закінчення таблиці 4.19

2. Товар у реальному виконанні	Технічна підтримка	Нм	Тх
	Приватність	Нм	Тх
	Налаштування під користувача	Нм	Ор
	Якість: Документація, яка допоможе правильно використовувати систему		
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент.			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товарианалоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
~300 000 грн./од.	~500 000 грн./од	~10 000 000 грн./од	~300 - 300 000 грн./од

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.22):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Замовлення специфічної реалізації, або використання існуючої, пробний термін, оплата.	Надання програмного забезпечення для розуміння вільної української мови з метою отримання певних послуг.	Канал збуту планується у межах взаємодії між замовником на виробником. Також, можуть надаватися обмежені версії продукту напрямую користувачам з метою реклами.	Реалізація декількох бета-версій для відображення загальних можливостей. Реалізація під конкретні вимоги замовника.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Підприємницька діяльність та сфера послуг.	Веб-сайт виробника.	Зручність, доступність, комфорт.	Донесення переваг до клієнтів.	Показати всі переваги

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі

товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки до розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

ВИСНОВКИ

У ході виконання магістерської роботи було описано актуальність теми, що розглядається, описана проблематика та необхідність її вирішення. У першому розділі також були описані існуючі типи діалогових систем. Був проведений аналіз особливостей, характерних для інтелектуальних помічників. Наведено багато прикладів існуючих рішень для кожного типу агентів. Серед яких були представники основані на командах (Mondly, Buoy Health Chatbot, Realbot Chatbot), на нейронних мережах (Casper, Roof AI, Mya Chatbot) та особливі реалізації, такі як Visual Dialog, Hipmunk, UNICEF. Розглянуті підзоди до побудови інтелектуальних агентів. На основі проведеного аналізу сформовані вимоги до розроблюваної системи.

У другому розділі розглядалася інформаційна модель. Детально описана функціональна модель її принцип функціонування, та основні частини: NLU та Core. Присутня інформація про підготовку даних. Надаються інформація про процес, відомий як word embedding, так як він є дуже важливою частиною, для функціонування розробленої системи. Розглянута нейронна мережа, яка є основою програмного забезпечення, із зазначення відмінностей між звичайною рекурентною мережею та на основі LSTM блоків, архітектура яких також була описано. Обґрунтовано вибір та причини використання цієї моделі. Надані відомості про процес навчання.

У третьому розділі мова йшла про архітектурне рішення для програмного забезпечення. Описана схема покрокового алгоритмічного функціонування. Розглянуто процес взаємодію користувача за додатком та розібраний інтерфейс. Наведена інформація про інструменти, які використовувалися, з аргументацією доцільності та оптимальності їх використання. Сформовані вимоги до технічного та програмного забезпечення. Надані дані про дослідницьку діяльність та її результати.

У четвертому розділі був проведений повний маркетинговий аналіз стартап-проекту. У ході якого було визначено новизну продукту, доцільність реалізації, здійснений аналіз ринку і конкурентів, і як результат, сформовані унікальні аспекти товару. Досліджений ринок на можливість входження. Обрана базова стратегія та стратегія конкурентної поведінки. Розроблений план по впровадженню проекту на сучасний ринок.

У результаті всієї роботи був розроблений перший інтелектуальний агент на основі української мови з можливістю взаємодії у вільному форматі. Вдосконалений процес попередньої обробки даних, до входження в нейронну мережу, що покращило точність відповідей і розуміння контексту розмови. За рахунок використання особливої структури на визначенні тематики, до якої відноситься повідомлення, та word embedding моделі bag of words вдалось досягти дуже швидкої взаємодії між користувачем на програмно-апаратним забезпеченням. Завдяки мінімалістичному інтерфейсу був досягнутий високий рівень комфорту, при використанні додатку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.
2. Paramveer Dhillon, Dean Foster and Lyle Ungar, Eigenwords: Spectral Word Embeddings, Journal of Machine Learning Research (JMLR), 16, 2015.
3. T. Joachims, T. Finley, Chun-Nam Yu, Cutting-Plane Training of Structural SVMs, Machine Learning, 77(1):27-59, 2009.
4. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

ДОДАТКИ

ДОДАТОК А

Демонстраційні плакати

Загальна модель роботи

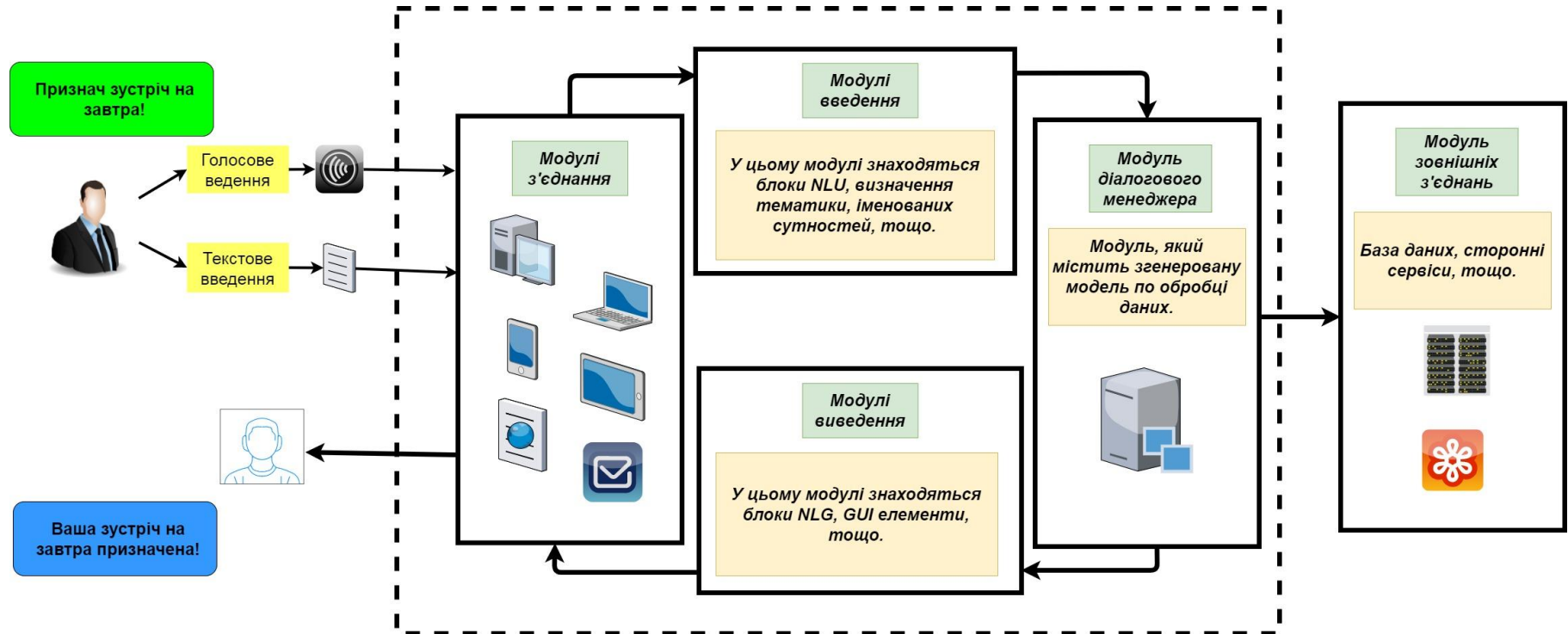
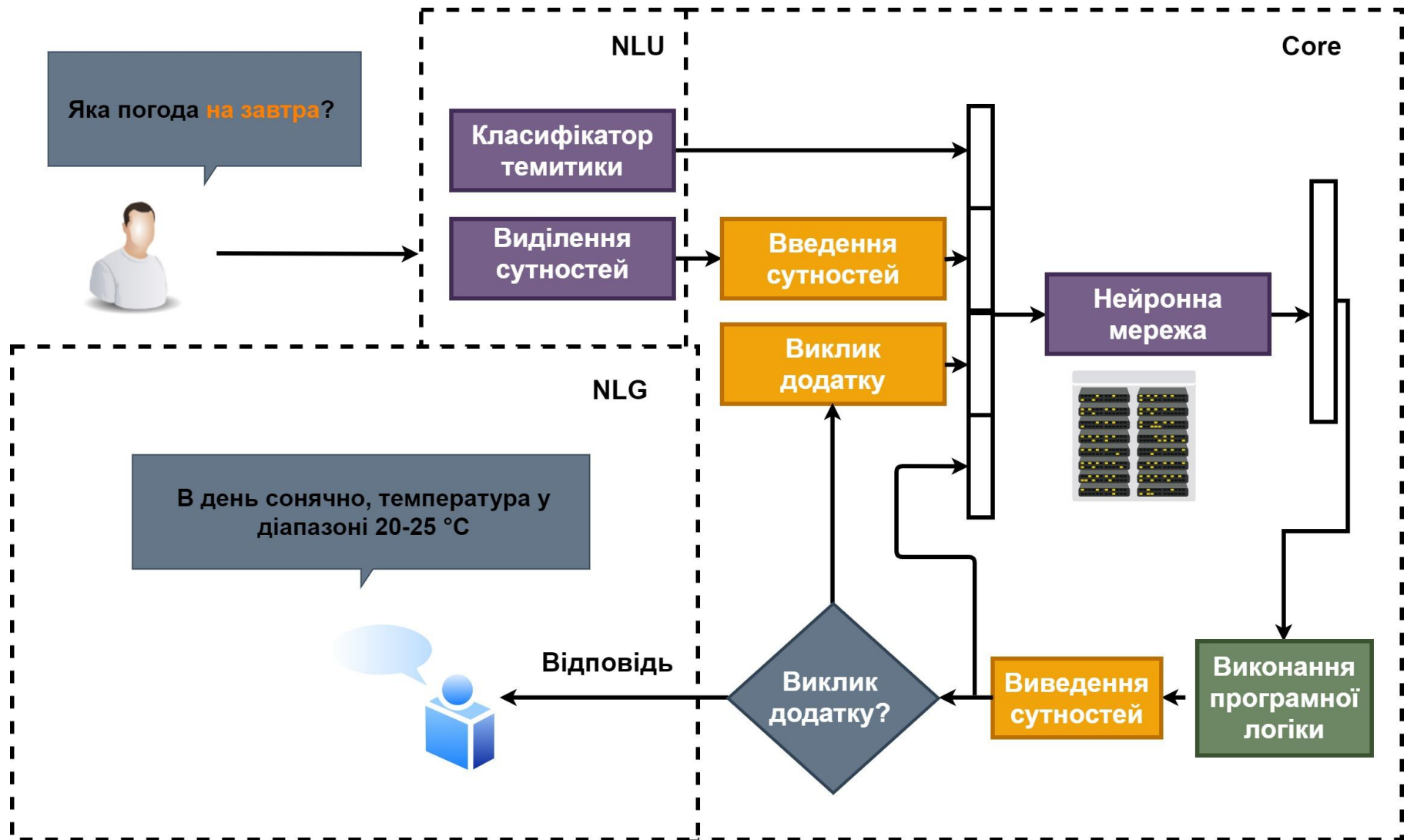
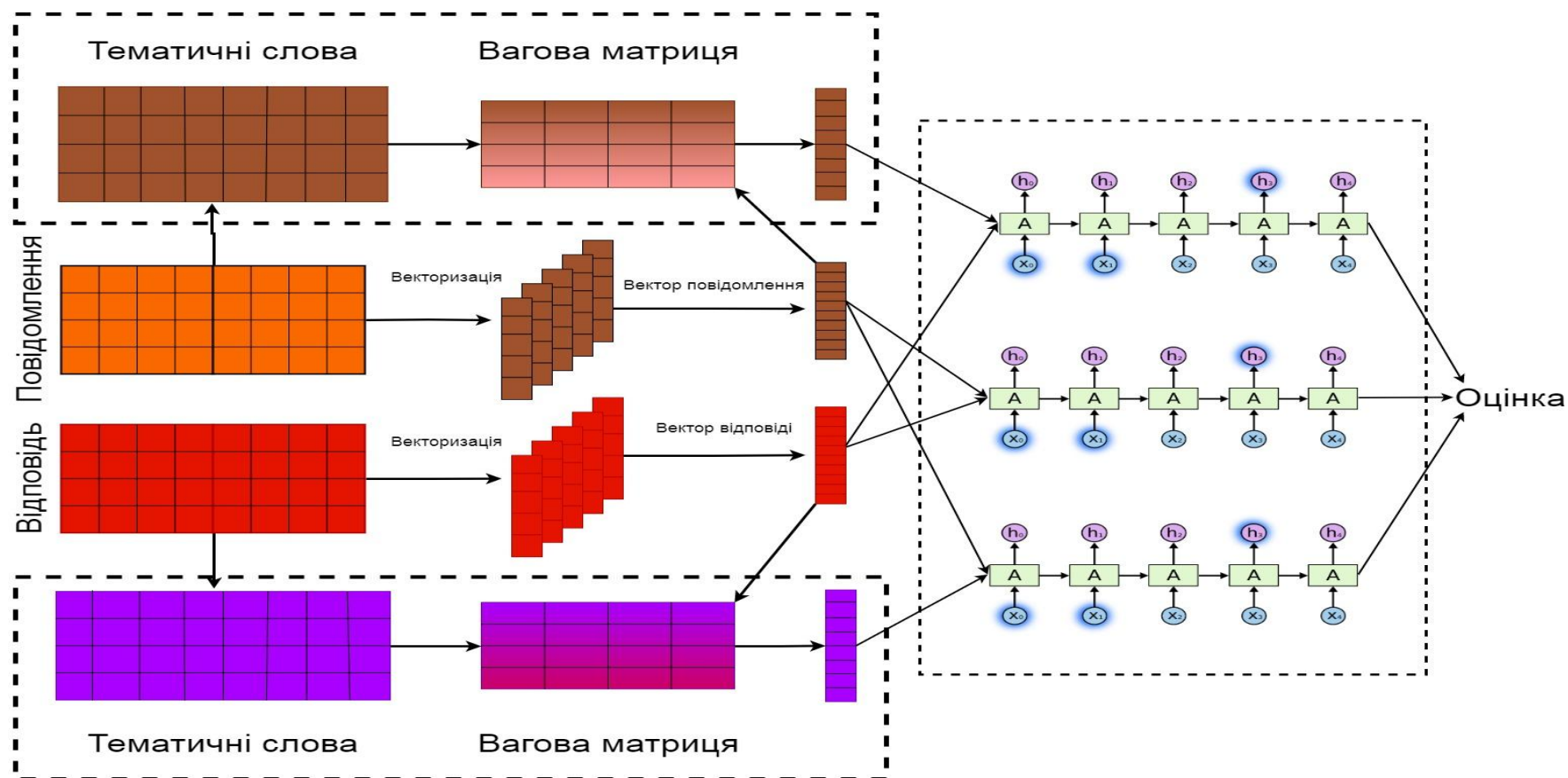


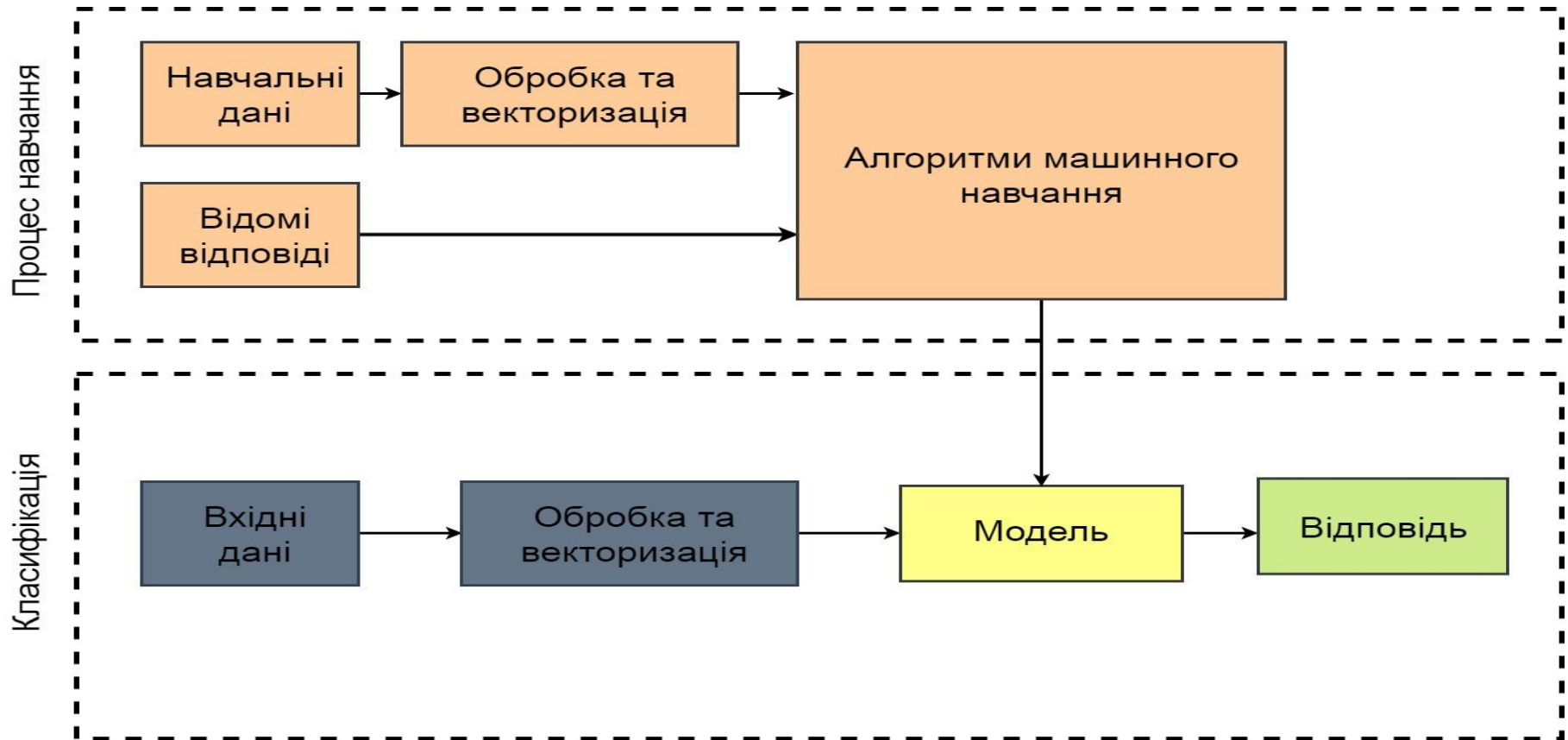
Схема внутрішньої роботи додатку



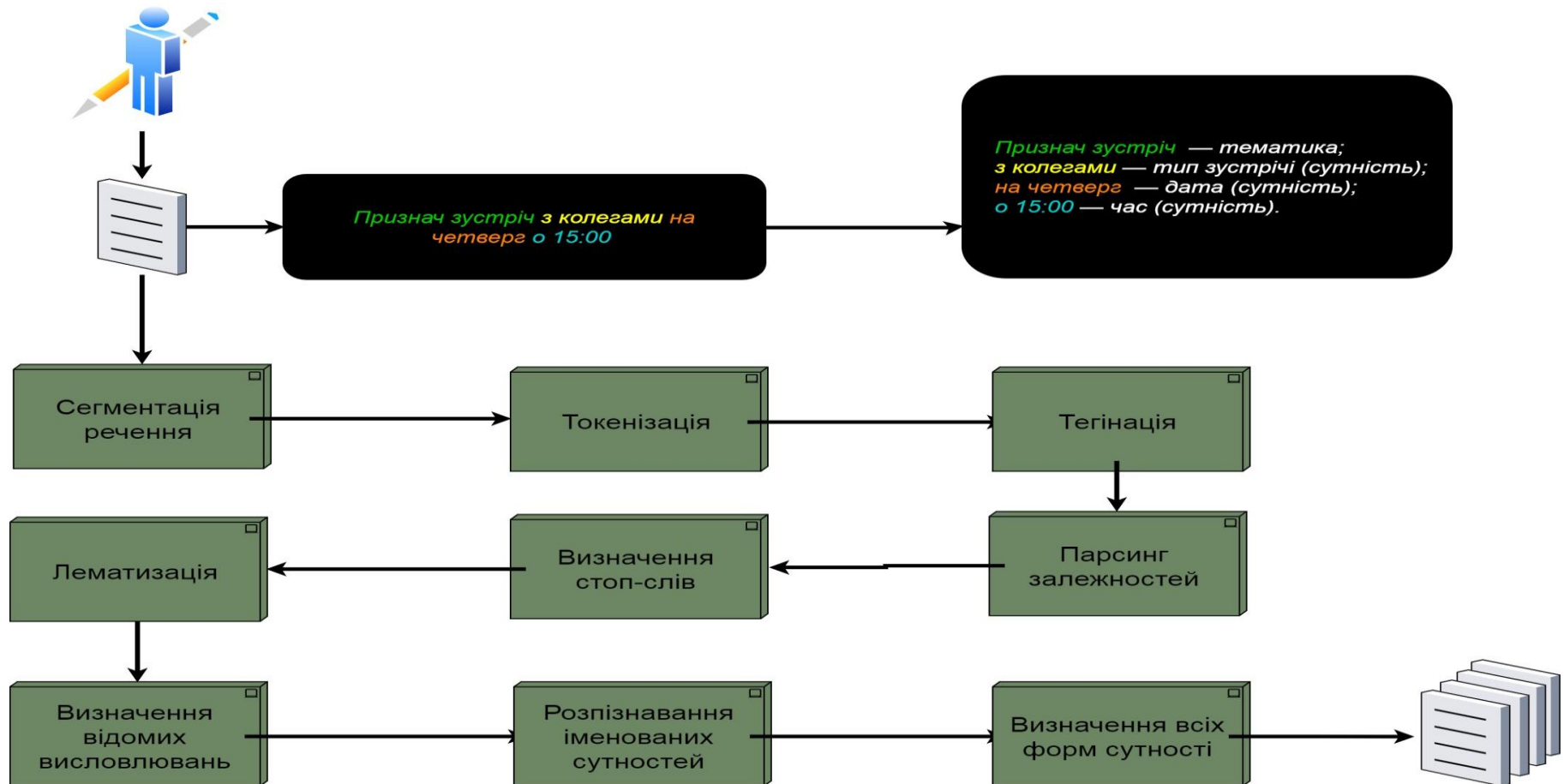
Структура нейронної мережі



Процес навчання та класифікації

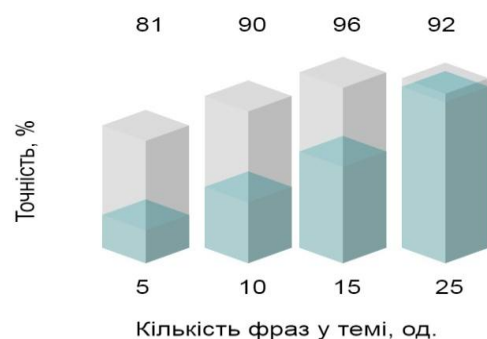


Механізм аналізу повідомлення

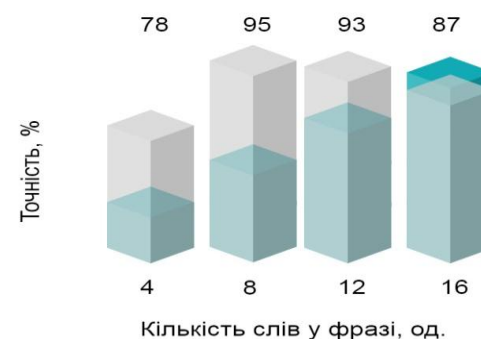


Результати проведеного дослідження

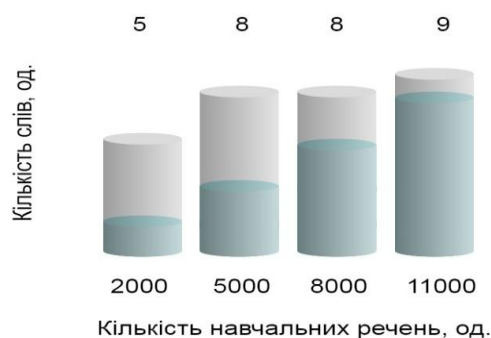
Залежність точності від кількості навчальних фраз у темі



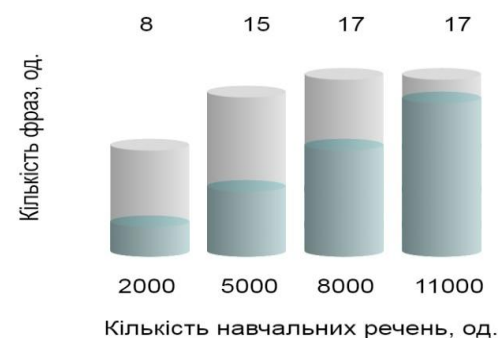
Залежність точності від кількості слів у фразі



Залежність кількості слів, які є оптимальними для однієї фрази при зміні об'єму даних для навчання



Залежність зміни кількості фраз при зміні об'єму даних для навчання



ДОДАТОК Б

Перевірка на співпадіння